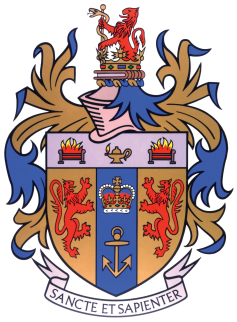
Optimisation-based methodologies for complex data analysis

Silva, Jonathan Cardoso

Awarding institution:
King's College London

Optimisation-based methodologies for complex data analysis



Jonathan Cardoso Silva

Supervisor: Dr. Sophia Tsoka

Department of Informatics

King's College London

This dissertation is submitted for the degree of

Doctor of Philosophy

November 2018

I would like to dedicate this thesis to my loving mother.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text. This dissertation contains fewer than 150,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Results presented in *Chapter 3: Sequential Clustering of Dynamic Networks* are a product of collaborations with Dr. Lazaros G. Papageorgiou, Chemical Engineering Department at University College London, and Dr. Laura Bennett, Department of Informatics at King's College London.

The work leading to this thesis and the publications contained herein was supported by the Coordenação de Pessoal de Nível Superior (CAPES), Brazil (Process number 13312138).

Jonathan Cardoso Silva

November 2018

Acknowledgements

I would like to thank my supervisor, Dr. Sophia Tsoka, my second supervisor Dr. Nishanth Sastry and to Prof. Lazaros Papageorgiou, for their encouragement and support during my PhD and for our many fruitful and inspiring conversations.

I am also very thankful to my colleagues Dr. Laura Bennett and Dr. Lingjian Yang from UCL with whom I had the opportunity to work in research projects together.

Many thanks also to all my dear friends at King's. I am really grateful to have been part of such a supportive, friendly and welcoming community. Special thanks goes to my friends in the lab, Dr. Gareth Muirhead and Lukas Diekmann.

Financial support from Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil, is also gratefully acknowledged.

Many thanks also to George Papadatos for introducing me to the topic of drug discovery from a computational perspective introducing me to interesting open source tools and projects.

Finally, I would like to thank my family and friends in Brazil, who were always present during my PhD even though they were on the other side of the ocean. In particular, many thanks to my mom, my grandmothers and Janquiel for their immense and constant support.

Abstract

Networks are a natural representation of data collected across many disciplines. The complex relationships between entities studied in these fields, whether it be people, computers or molecules, cannot be fully characterised individually but are, instead, better described by computational models as a function of their overall interactions. This thesis focuses on the development of such models to detect communities and to predict outcome variables in real networks using mathematical programming, a transparent, flexible and customisable modelling paradigm.

First, the temporal evolution of groups in complex social networks is explored. Various methods detect groups in dynamic networks either by aggregating all temporal contact information into a single network or by looking at snapshots of time independently. In this work, a more robust approach is employed where, at each time step, both the current and previous modular structures of a network are considered. A mixed integer non-linear programming (MINLP) model is proposed to capture more stable patterns of change and was shown to match the ground truth of networks.

Next, the development of Quantitative Structure-Activity Relationship models (QSAR) is addressed. These regression models are vastly used in drug discovery and aim to predict biological activity from the attributes of molecules. In this work, algorithms are proposed to divide the compounds in sub-groups either by

their molecular features or from modules that naturally arise when representing this data as a network. Suitable equations that predict biological activity are then identified for each group by a mathematical programming model. These algorithms create predictive, customisable and interpretable QSAR sub-models, which can later be used for virtual screening, SAR studies or lead optimisation of drug candidates.

Overall, this thesis proposes computational models to optimisation problems in regression and network analysis. The proposed methods produce transparent and interpretable solutions towards a better understanding of the dynamics of social systems and have the potential to assist in the endeavours of drug discovery.

Publications

During the compilation of this thesis, the following related articles have been published by the author:

- **Jonathan C. Silva**, Laura Bennett, Lazaros G. Papageorgiou, and Sophia Tsoka. A mathematical programming approach for sequential clustering of dynamic networks. *The European Physical Journal B, 89(2):39*, feb 2016. DOI: 10.1140/epjb/e2015-60656-5.

- Lingjian Yang, **Jonathan C. Silva**, Lazaros G. Papageorgiou, and Sophia Tsoka. Community Structure Detection for Directed Networks through Modularity Optimisation. *Algorithms*, 9(4):73, 2016. DOI: 10.3390/A9040073.

At the time of the submission of this thesis, the following manuscripts are being prepared for publication:

- **Jonathan C. Silva**, Lazaros G. Papageorgiou, and Sophia Tsoka. Optimal Piecewise Linear Regression algorithm for QSAR Modelling. **(in preparation)**

- **Jonathan C. Silva**, Lazaros G. Papageorgiou, and Sophia Tsoka. Hierarchical network-based regression algorithm for QSAR Modelling. **(in preparation)**

Work related to this research project has also been disseminated in the following conferences:

- **Jonathan C. Silva**, Laura Bennett, Lazaros G. Papageorgiou, and Sophia Tsoka. Sequential Clustering of Dynamic Network Snapshots using Mathematical Programming. *NetSci,International Conference on Network Science*, 2015.

- **Jonathan C. Silva**, Lazaros G. Papageorgiou, and Sophia Tsoka. Model tree by ensemble of piecewise linear models and its application to QSAR modelling. *ISMB/ECCB, International Society for Computational Biology*, 2017.

- **Jonathan C. Silva**, Laura Bennet, Aristotelis Kittas, Linjian Yang, Sonsong Liu, Lazaros G. Papageorgiou, Sophia Tsoka. Community detection algorithms for analysis of biological networks. *Informatics for Health 2017*, 2017.

Table of contents

List of figures **xiv**

List of tables **xviii**

1 Introduction **1**

1.1 Overview 1

1.2 Research aims 3

1.3 Thesis outline 4

2 Background **6**

2.1 Complex Networks 7

2.1.1 Properties of real networks 9

2.1.2 Modularity optimisation 12

2.2 Mathematical Programming 17

2.2.1 Mathematical programming formulations of modularity metric 19

2.2.2 Alternative metrics for community detection and mathematical programming formulations 21

2.2.3 Comparison between community detection metrics 24

2.3 Quantitative Structure-Activity Relationship (QSAR) 27

2.3.1 Introduction to QSAR models 28

2.3.2 Molecular Descriptors 30

2.3.3 Validation procedures 32

2.3.4 Molecular Similarity 33

2.3.5 Activity cliffs 35

2.4 Summary 37

3 Sequential Clustering of Dynamic Networks 38

3.1 Introduction 40

3.2 Methods 43

3.2.1 A mathematical programming model for sequentially clustering snapshots of dynamic networks 43

3.2.2 Network data 47

3.2.3 Alternative clustering methods 49

3.2.4 Adjusted Rand Index 49

3.3 Results and discussion 50

3.3.1 Synthetic network 50

3.3.2 High School network 52

3.3.3 MIT Social Evolution dataset 54

3.3.4 Brazilian Congress voting dataset 56

3.4 Comparative analysis 57

3.5 Conclusions 61

4 Regression algorithms for QSAR models 65

4.1 Introduction 66

4.1.1 Data Sets 68

4.1.2 New mixed integer programming model 72

4.1.3 Proposed algorithm 75

4.1.4 Implementation and Validation scheme 77

4.1.5 Comparative analysis 77

4.2 Results and Discussion 77

4.2.1 Parameter optimisation 79

4.2.2 Algorithm results 80

4.2.3 Overall Variable Importance 83

4.2.4 Custom constraints to the model 85

4.2.5 Comparison with other algorithms 87

4.3 Conclusions 89

5 Predictive QSAR models incorporating chemical networks 91

5.1 Background 92

5.2 Network analysis 94

5.2.1 Network construction 94

5.2.2 Presence of activity cliffs 96

5.2.3 Structural properties of modules 98

5.3 Network algorithm 102

5.3.1 Prediction of new samples 105

5.3.2 Implementation details and algorithm validation 105

5.4 Results and Discussion 106

5.4.1 Overall performance 106

5.4.2 Neuropeptide Y inhibitors 109

5.4.3 CHRM3 112

5.4.4 DHFR inhibitors 115

5.4.5 Robustness of modules 118

5.4.6 Improvement over piecewise algorithm 119

5.4.7 Generating constraints for *de novo* molecular design 121

5.4.8 Comparative Analysis 123

5.4.9 Enforcing a minimum number of neighbours 124

6 Conclusions and future work **131**

6.1 Concluding Remarks 131

6.2 Future work 133

References **136**

List of figures

2.1 Example of adjacency matrix of an unweighted undirected network 8

2.2 Visualisation of network represented by its adjacency matrix . . . 8

2.3 Communities corresponding to maximum modularity in the karate network 13

2.4 Performance of community detection metrics for synthetic networks. 26

2.5 Modules in synthetic network identified by modularity 26

2.6 Modules in synthetic network identified by ratio association . . . 27

2.7 2D Structure of acetaminophen 30

2.8 Example of network representing similarity between molecules and activity cliffs 36

3.1 Example of dynamic synthetic network 50

3.2 Partitions detected by SeqMod for the synthetic network, using $\varepsilon = 0.00$. 51

3.3 Partitions detected by SeqMod for the synthetic network, using $\varepsilon \geq 0.10$. 52

3.4 High School network - The change in Adjusted Rand Index (ARI) at each time step for SeqMod depending on the value of ε. 53

3.5 Partitions detected for High School network by SeqMod, with $\varepsilon = 0.15$. 54

3.6 MIT Social Evolution network - The change in Adjusted Rand Index (ARI) at each time step for SeqMod depending on the value of ε. 56

3.7 Brazilian congress network - The change in Adjusted Rand Index (ARI) at each time step for SeqMod depending on the value of ε. 56

3.8 High School network - The change in Adjusted Rand Index (ARI) at each time step for each algorithm. 59

3.9 MIT Social Evolution network - The change in Adjusted Rand Index (ARI) at each time step for each algorithm. 59

3.10 Brazilian Congress network - The change in Adjusted Rand Index (ARI) at each time step for each algorithm. 59

4.1 Distribution of activity, measured by the logarithm of the 50% inhibitory concentration (IC_{50}) of compounds in the studied data sets. 70

4.2 Validation scheme adopted in this study. 78

4.3 Breakpoints, regions and equations found by OPLRAreg for data set hDHFR 82

4.4 Breakpoints, regions and equations found by OPLRAreg for data set NPYR1 82

4.5 Piecewise model for hDHFR inhibitors with khs.aaNH as the partition feature 86

4.6 Comparison of OPLRAreg to other machine learning algorithms . 88

4.7 Comparison of OPLRAreg to other machine learning algorithms (Dataset rDHFR) 88

5.1 Threshold analysis for network representation of QSAR datasets used in the study 95

5.2 Number of singletons for each data set according to threshold level in the interval $0.15 \leq t_\alpha \leq 0.30$ 96

5.3 Examples of network visualisations generated with optimal thresholds 97

5.4 Activity cliffs in NPYR1 network 99

5.5 Common structure of molecules in module m05 of rDHFR network 99

5.6 Module-assay correspondence for network rDHFR 100

5.7 Maximum common structure of each module in rDHFR network . 101

5.8 Most common rings and frameworks for modules m01 and m08 in rDHFR network 102

5.9 Modular (OPLRA) algorithm 104

5.10 Low activity cliffs and interquartile range of prediction errors in the external set 109

5.11 NPYR1 network modules and predictive performance in the external set 110

5.12 NPYR2 network modules and predictive performance in the external set 111

5.13 CHRM3 network modules and predictive performance in the external set 114

5.14 Maximum common substructures of samples m06 in CHRM3 network 115

5.15 hDHFR network modules and predictive performance in the external set 116

5.16 rDHFR network modules and predictive performance in the external set 117

5.17 Frequency of co-occurrence of nodes in the same module averaged for 100 sub-sampled networks. 118

5.18 Regions identified by OPLRAreg for dataset rDHFR 120

5.19 Maximum common substructures of r01 and r02 120

5.20 Fragment represented by descriptor khs.aaN 122

5.21 Comparative results of machine learning algorithms in QSAR data sets 124

5.22 Examples of network visualisations generated with optimal thresholds 126

List of tables

2.1 Examples of molecular descriptors calculated for acetaminophen . 31

3.1 Summary of networks used in this paper. 47

3.2 Best results found by each algorithm and the respective parameters that has returned such results. 58

4.1 Summary of data sets used in this study, comprising all samples tested as inhibitors against a common drug target 69

4.2 Performance of piecewise linear algorithm for different regularisation parameters. 79

4.3 Average number of regions and selected features found by OPLRAreg during cross-validation 80

4.4 Top 15 features and their importance score for each data set . . . 83

5.1 Optimal threshold values and network metrics of QSAR data sets 95

5.2 Proportion of activity cliff classes in the QSAR data sets studied. 98

5.3 Performance of Modular (OPLRA): average mean absolute error and deviation in the cross-validation 107

5.4 Average performance of Modular (OPLRA) in the external set . . 107

5.5 Reduction in MAE and SD of errors by Modular (OPLRA) compared to OPLRAreg per discontinuity class 121

5.6 Metrics for QSAR networks with a minimum number of neighbours $k = 3$ 127

5.7 Performance of Modular (OPLRA) with $k = 3$ in the external set 127

Chapter 1

Introduction

1.1 Overview

The large volume of data collected, generated and stored in computers and databases increases continuously, powered by advances in technology and Internet infrastructure [1, 2]. The flow of data across the Internet grew more than 96 million times since 1990, reaching a record traffic in 2016 with a flow of 1.2 zettabytes. This number is set to triple in five years [3].

This process has not only facilitated the access to information but it has also changed our daily routines and shaped our social interactions. If online retailers know what we are likely to buy next, it is because they have collected enough data describing the behaviour of thousands of other customers in a similar context and have trained a computer model capable of making predictions about our specific shopping experience. Similarly, social media websites and the advertisement industry have been taking advantage of our digital traces online, our friendships, tastes, likes and dislikes, to curate what we are going to read and see and to offer

targeted promotions. The "Big Data" era has also made its impact in science; enabling seamless collaboration among researchers across the globe and facilitating the spread of findings to a wider audience [4].

Machine Learning (ML) was born at the heart of this data revolution. The discipline combines elements from statistics, computer science and artificial intelligence and deals with algorithms to identify relationship patterns and make predictions based on what it has learned from data [5, 6]. The learning process varies from algorithm to algorithm but the main components are almost always the same. Raw data must first be processed and put in a structured format, usually in tabular form where each row represents a single observation of data and columns represent the attributes/features/descriptors of the data. The machine will then be trained to combine these features and develop a general description of the data.

In a supervised learning setting, the goal is to establish a relationship between an independent variable, or outcome, to features in the data set. The models are called regression when the outcome assumes numerical values such as price of objects, scores, experimental results, and classification when modelling categorical variables, such as yes/no, colours, classes and groups. The resulting model can then predict the outcome of unseen samples which have not been used to train the algorithm. In an unsupervised context, the outcome is not provided or not known and the objective is to find clusters or other arrangements based only on the structural properties of data. With the help of mathematical formulas, heuristics and even some random processes, the model will then try to find the solution that maximises some predefined performance metric.

Even in situations when relationships could be spotted by a human, machine learning makes the process faster, simpler and automatic [7]. The prominence of

this automation process raises some questions about the transparency, or the lack thereof, in many algorithms. Take as an example, the artificial neural networks or deep learning, as it is more frequently referred to nowadays, a technique inspired by the brain where the weights of its artificial neurons are trained to model the input data as it propagates through its layers. Deep learning has excelled in image classification [8] and speech recognition [9] but it has also produced predictive models in many other areas of research [10]. But even though these carefully trained networks can generalise well to new and unseen data, the artificial neurons are unable to explain which attributes of the data have lead to its final output [11].

The development of transparent models has started to attract the attention of the machine learning community [12–16] and is one of the main topics of the research developed and described in this thesis. The principles and models presented herein are applicable to areas such as drug discovery and bioinformatics where the correlation between features and outcome variables is not always well understood. Interpretability is tackled mainly by the choice of modelling paradigm. Here, mathematical programming is used as the default framework to represent and solve optimisation problems. This technique is useful when exact and optimal solutions are preferred over approximations and is widely used in operational research, logistics and engineering.

1.2 Research aims

The overall aim of this research is to develop optimisation models using mathematical programming to solve regression and clustering problems in networks. The interdisciplinary nature of this work allowed for the application of these models

in social sciences and in cheminformatics, with contributions to drug discovery. More specifically, the main research goals of this thesis are:

- Extend modularity optimisation algorithms to community detection incorporating temporal information in dynamic networks
- Develop predictive Quantitative-Structure Activity Relationship (QSAR) regression models using mathematical programming
- Explore community detection in network representations of bioactivity data
- Combine complex network analysis and regression algorithms to build interpretable QSAR models

1.3 Thesis outline

The rest of this thesis is organised as follows.

Chapter 2 introduces the concepts used in the thesis. A background in optimisation techniques, network analysis, community detection, regression and quantitative structure-activity relationship is presented.

Chapter 3 addresses the problem of identifying clusters in dynamic networks. The problem is formulated as a mathematical programming model that takes into consideration the connectivity history in the network to more accurately identify the evolution of groups. Networks of social contacts and political similarities were used to demonstrate the capabilities of the proposed method.

In Chapter 4, the problem of developing quantitative structure-activity relationships (QSAR) models is addressed. An optimal piecewise linear regression algorithm was modified to successfully create QSAR models for five data sets of

protein inhibitors. The algorithm clearly identifies rules to separate the data and linear equations to fit the data in each distinct group.

Chapter 5 presents a new methodology that combines network analysis and the piecewise model introduced in Chapter 4 to create QSAR sub-models.

The methodology chapters (Chapters 3-5) are structured as independent research questions. In fact, Chapter 3 has already been published as a journal paper at the European Physics Journal B while the content of Chapters 4 and 5 are being prepared for consideration in other suitable peer-reviewed journals.

Chapter 2

Background

The main topics in this thesis are the detection of groups in networks and the development of regression models, applied to temporal network analysis and for the development of predictive models for drug discovery. This chapter reviews essential concepts, methods and work related to these topics and paves the way for the proposed methodologies and algorithms presented in the remaining chapters. This includes an introduction to concepts of network analysis, the main metrics, topological properties of networks and methods for community detection. Next, the area of Quantitative Structure-Activity Relationship (QSAR) models is presented, introducing concepts of molecular descriptors, activity cliffs and biological activity prediction. Finally, the formalism of optimisation problems is presented, along with a brief description of the different types of mathematical programming models and its solving techniques.

2.1 Complex Networks

The Internet, the World-Wide-Web, on-line social networks, food webs, protein interactions and metabolic processes in the body are examples of networks. These interconnected systems are made of individual agents that interact with each other in an organised way and give rise to a number of patterns and topological properties. Networks have been the object of study of Graph Theory, a branch of discrete mathematics, since the eighteenth century but have received a renewed attention from a wider community recently due to its social, technological and scientific applications in the modern world [17, 18]. Propelled by the boom of the Internet and the easier access to large volumes of data, network research has shifted its focus from the study of small graphs to the statistical properties of larger and more complex networks [19].

A network, or a graph, G is a mathematical structure consisting of a set of items N, also called *nodes* or *vertices*, which are associated by *edges* or *links*. In its simplest form, a graph can be represented by an adjacency matrix $\mathbf{A}$, a symmetric $|N| \times |N|$ binary matrix indicating the connectivity between the nodes. Whenever a node i is connected to j, $A_{ij} = 1$; otherwise, $A_{ij} = 0$. An example of an adjacency matrix and the visualisation of its correspondent graph are shown in Figures 2.1 and 2.2, respectively.

In this example, the network is unweighted and undirected, nodes are just simply either connected or not, but a network can encode much more information about the relationships between its nodes. For example, a node i may be connected to j but there might not exist an edge from j that points to i. In this case, the adjacency matrix is no longer symmetric and we say that the network is directed. Metabolic networks are examples of directed networks where the order of reactions is naturally represented by directed links. Edges might also have weights, numerical

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 2.1 Example of adjacency matrix of an unweighted undirected network

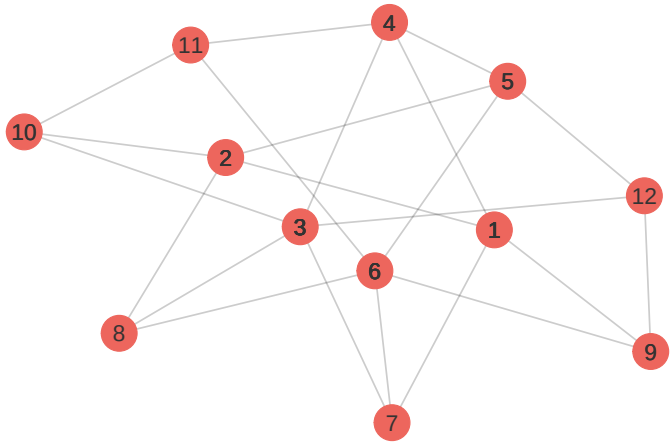


Fig. 2.2 Visualisation of network represented by adjacency matrix of Figure 2.1

values indicating the strength, distance or similarity between nodes (weighted network) or contain labels indicating the type of connection between vertices.

The presence and strength of connections may also change over time. Examples of temporal, or dynamic, networks are person-to-person communication and contact, information broadcast, brain and cell networks [20–22]. In this type of network, a sequence of adjacency matrices represent network states at different time slices and the selection of time window is application dependent. Suppose, for example, that we want to model phone calls over a period of time as a network. If we want investigate the different dynamics of day versus night calls, we would

probably represent the network in hourly slices or we could divide it by periods of the day. If instead, we are interested in the peaks of phone calls around holidays, we might want to look at the calls made per week or month. We can also choose whether to represent temporal networks as weighted versus unweighted, directed versus undirected. Temporal networks are one specific realisation of multilayer networks, a generic network representation that have been receiving a lot of attention recently and represent different layers of relationships between nodes [23–25].

2.1.1 Properties of real networks

Networks can be characterised by their statistical properties. The *degree* d_i of a node i indicates its number of neighbours while the *average degree* characterises the connectivity of the entire network. The *edge density* represents the number of edges divided by the number of all possible edges in the network, nodes in a network with high edge density are more interconnected than the sparse connections observed in graphs with low edge density. Another metric, *average shortest paths* or *path length*, represents the number of edges in the shortest path between two nodes averaged over all node pairs in the network. This metric indicates how compact the graph is, signals propagate quicker in a network where the distances between nodes is smaller (small path length).

Another metric that helps characterize the density of connections in a network is the *transitivity* or *clustering coefficient* [19], a property that relates to the probability that nodes in the same neighbourhood are connected to each other. If nodes j and k are connected to i, it is likely that j and k are also connected together, which forms a triad around i: i, j, k. The clustering coefficient of i (C_i) represents the number of such triads that exist involving i compared to the

number of triads that would be formed if all its neighbours were interconnected. Mathematically, the formula of C_i is expressed as below:

$$C_i = \frac{t_i}{(d_i(d_i - 1))/2}, \tag{2.1}$$

where t_i represents number of triads in which node i participates and the denominator represents the total number of triads in which i could participate. When $C_i = 0$, neighbours of node i are not connected and when $C_i = 1$, all neighbours of i are connected.

It is also common to represent the global transitivity of the network to numerically characterize the tendency that all nodes in the network are strongly connected in their own neighbourhoods. One way to measure global transitivity is by average clustering coefficient (ACC), given by the simple average of C_i for all nodes in the network:

$$ACC = \frac{1}{|N|} \sum_{i}^{N} C_i. \tag{2.2}$$

Note that a network with low ACC values (close to zero) can become disconnected easily [26, 27]. Imagine for example a network of nodes placed on a line so that the connections are only between neighbours located immediately to the left and right of each node. If any link of this network is removed, the network will disintegrate into two components, two sub-graphs. However, many real networks tend to have high ACC values. In these networks, it is usually also possible to observe the existence of cohesive groups since the neighbourhoods are well connected. This effect is also popularly known as "six degrees of separation" [28–30] in reference to the work of Travers and Milgram in the 1960s where it

was proposed that every person in the planet can be connected to anyone else by only six acquaintances [31]. In food webs, the degree of separation is even lower, species are separated on average by two links [32].

The scale-free property is another common feature of real networks and is related to the way networks naturally evolve [33]. As a network grows, new nodes tend to link to highly connected nodes, in detriment of those with few neighbours. This preferential attachment causes node degrees in scale-free networks to follow a power-law distribution and, as a result, few *hub* vertices have a large number of neighbours while the majority of nodes have a small degree [34]. Hub nodes are involved in the convergence of opinions in social media [35], associated with proteins encoded by essential genes in protein-protein interaction networks [36] and also play an important role in the resilience of network architecture [37].

Another important property of real networks is the presence of community structure [38, 39]. Nodes that share some sort of similarity tend to group and have more links between them than with the rest of the graph. Communities contain valuable information about the function and organization of a network and its identification leads to a better understanding of the nature of the observed system. By detecting communities, we can identify aggregations in a social network, a group of similar pages in the Web and link topological groups of proteins to functional features, to name a few examples.

The intuition behind groups in a network is easy to grasp but community detection is a difficult problem to define and solve mathematically. We can identify communities by minimizing the number of edges between partitions of the graph [40], or we could think of a community as a subgraph where the number of internal connections is larger than the number of edges pointing out of the subgraph [41]. Another approach is to model communities as stochastic block models [42],

representing the probability that vertices share an edge in a subgraph. Regardless of the definition of a group or module, community detection can be seen as an optimisation problem. Most popular community detection techniques define an objective function which represents the quality of partitions and an algorithm is implemented to detect the partition corresponding to the best value of that objective function.

A measure called modularity is the most widely used quality function for community detection and is the metric used in this thesis to define and detect modules. An introduction to modularity optimisation is given below.

2.1.2 Modularity optimisation

Modularity (Q) was introduced in [43] as a measure of how communities in a real network deviate from the connectivity observed in a random graph and is related to a property called assortativity. An assortative network contains a high fraction of edges running between vertices of the same group. In random models where there is no formation of groups, this fraction is smaller [44]. Therefore, it is possible to quantify the modular structure of a network by subtracting the fraction of edges expected to occur if they were located at random from the fraction of edges actually present in a module of the network.

Modularity can assume negative values when each node is considered as a different community, indicating that the graph has no community structure [38]. It has value zero when the whole graph is considered as a single community and it increases with higher number of internal edges inside communities. The best partitions are the ones with maximum modularity so, the problem of detecting communities becomes the maximization of modularity.

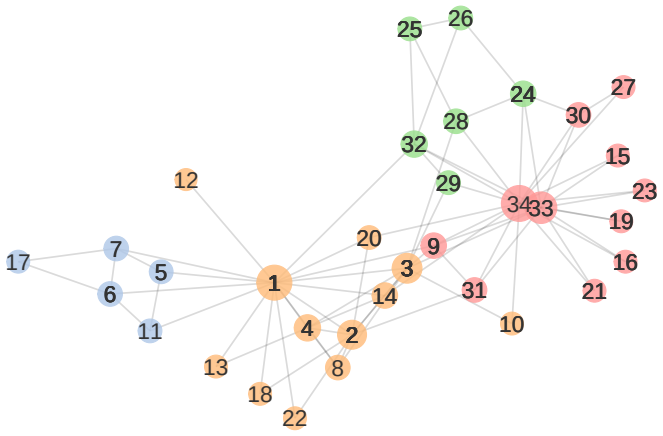


Fig. 2.3 Karate network and communities corresponding to the maximum modularity value

Figure 2.3 illustrates the partition corresponding to the maximum modularity value ($Q = 0.4188$) in the famous Zachary's karate network [45]. Colours indicate the communities and the size of nodes vary with their degrees. The network represents the relationship between members of a karate club. A conflict between the coach (node 1) and the owner (node 34) over the price of karate lessons divided the club, creating the group divisions we see in the graph.

The formula of modularity is represented in Equation 2.3:

$$Q = \sum_{edges(i,j)} \delta(i,j) - P_{ij}, \tag{2.3}$$

where P_{ij} is the expected number of edges between two classes i and j and the first term of the equation represents the number of edges between vertices of the same community and is defined by:

$$\sum_{edges(i,j)} \delta(i,j) = \frac{1}{2} \sum_{ij} A_{ij} \delta(i,j), \tag{2.4}$$

where A_{ij} is the adjacency matrix of the graph and δ is the Kronecker delta:

$$\delta(r,s) = \begin{cases} 1 & \text{if } r = s \\ 0 & \text{otherwise.} \end{cases}$$

The random model considered in this equation is the configuration model [46, 47] and it consists of a graph generated by the same degree distribution of a real network. The generated graph contains a giant component without any communities [48]. The expected number of edges P_{ij} between two classes i and j in this model can be calculated [49] as follows.

Consider two vertices i and j with degrees k_i and k_j, respectively. The probability of having an edge from i, ending at j is $\frac{k_j}{2m}$, since the total number of incident edges in the network is $2m$, m being the total number of edges. Summing this probability over all edges of i we get $\frac{k_i k_j}{2m}$ and the expected number of edges between all pairs of vertices in the network that are in the same community is:

$$P_{ij} = \frac{1}{2} \sum_{ij} \frac{k_i k_j}{2m} \delta(i,j), \tag{2.5}$$

where the probability is multiplied by $\frac{1}{2}$ to prevent counting the same edge twice.

When we substitute P_{ij} of Equation 2.5 in Equation 2.3 and divide the terms by m to get the fraction, we obtain the following formula:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(i,j), \tag{2.6}$$

The equation above compares pairs of nodes to their expected value and can be easily adapted to weighted networks. In this case, A_{ij} is a numeric value that contains the edge weights between nodes i and j, and k_i indicates the strength (sum of weighted degrees) instead of the simple degree of a node.

Limitations of the modularity metric

Despite being the most popular optimization metric for community detection, modularity has its limitations. Fortunato and Barthélemy [50] have investigated the mathematical properties of the metric and proved that modularity has a resolution limit. Modules below the size limit $\sqrt{m}$, the square root of the number of edges in the network, might not be detected but are instead assigned to larger communities or in combinations of smaller weakly interconnected modules. These small communities are not detected by any algorithm that maximises modularity even when they are well defined.

The source of the resolution limit comes from the null model used to define modularity. A better metric should be able to identify the local interactions of the vertices instead of comparing the graph to a global model, in which every vertex is implicitly assumed to probably interact with every other vertex of the graph [38]. Several changes to the modularity formula have already been proposed to circumvent this limitation of the metric. The proposals involve the addition of a parameter to balance the two terms of equation 2.6 [51], the use of a different null model [52], or the introduction of link density in the objective function as in the case of the metric called "modularity density" [53]. However, while these alternatives are able to detect the small communities ignored by modularity, none of them solve the problem completely and may even create other unforeseen problems [39, 54].

Other limitation of modularity include its degeneracy i.e., it is possible to find multiple partitions for a network with equally optimal modularity values [55]. Also, modularity has been proved to be a NP-hard problem [56], the complexity of modularity optimisation grows more than exponentially with the size of the network and there are not any algorithms that can efficiently solve modularity to its optimality in very large networks.

Despite its limitations, modularity optimisation is still widely used for community detection [39] given that this metric is capable of providing useful and meaningful partitions for real networks. The most popular algorithm of modularity optimisation is a simple and fast algorithm named Louvain method, in reference to the University of Louvain, where this method was first developed [57]. Each node is initially assigned to a unique module; then, at each iteration, the pair of nodes leading to the maximum increase in relative modularity is merged and become a single point for the next iteration. The algorithm stops when it is not possible to improve modularity. Other algorithms include spectral methods [58], metaheuristics [59, 60] and simulated annealing [61].

Mathematical programming is another technique used to optimise modularity and detect optimal clusters for nodes in a network and is the optimisation method chosen to develop the models in this thesis. This technique usually identifies network modules more accurately than the methods mentioned above and allows for customisation of the optimisation process [62–66].

This thesis focuses on modularity metric and mathematical programming is the technique selected to identify modules in real networks applications. A brief description of this programming paradigm is given below, followed by its adapted formulation of modularity and other related metrics.

2.2 Mathematical Programming

Mathematical programming, or mathematical optimisation, is the process of identifying the best or, when that is not possible, good solutions to real-life optimisation problems [67]. These mathematical models are largely used in computer science, engineering and decision sciences for the optimisation of supply chain [68, 69], power generation [70], disease classification [71], smart grids [72, 73], air traffic management [74], telecommunication networks [75] and routing of vehicles [73].

An optimisation problem is formulated as follows:

$$\begin{aligned} \min_{\mathbf{x}}/\max \quad & f(\mathbf{x}) \\ \text{subject to} \quad & E(\mathbf{x}) = 0 \\ & I(\mathbf{x}) \leq 0 \\ & \mathbf{x} \in \mathbb{X} \end{aligned}$$

where $\mathbf{x}$ is the vector of decision variables in a subspace $\mathbb{X}$, $f(\mathbf{x})$ is the objective function to be maximised or minimised, and $E(\mathbf{x})$ and $I(\mathbf{x})$ are the equality and inequality constraints, respectively. The goal is to identify values for the decision variables that correspond to the best value of objective function, restricted by the variables subspace and the imposed constraints.

Optimisation models can assume different classes [76] according to the type of decision variables and objective function:

Linear Programming (LP): all decision variables are continuous and the objective function is linear

Non-linear Programming (NLP): decision variables are continuous and the objective function is non-linear (polynomial, trigonometric function)

Mixed Integer Linear Programming (MIP or MILP): contains discrete decision variables (binary, integer) and the objective function is linear

Mixed Integer Quadratic Programming (MIQCP): similar to MINLP but the non-linearities are quadratic

Mixed Integer Non-Linear Programming (MINLP): discrete decision variables and non-linear objective function

Optimisation problems are usually represented in a structured format and implemented in a modelling platform. Examples of traditional commercial platforms are GAMS [76], AMPL [77] and AIMMS [78] but nowadays optimisation packages can also be found in open source programming languages such as R [79] and Python with Pyomo [80]). Once the optimisation problem is represented, a solver is invoked to generate a solution to the problem, for example IBM CPLEX [81, 82], GUROBI [83], BARON, ANTIGONE, as well as the open source solvers found in Coin-OR project [84] such as GLPK and CBC.

These solvers identify solutions analytically using a combination of algorithms depending on the problem type. LP problems can be solved by the simplex algorithm, a method that starts from a initial solution and follows a path along the edges of the geometric representation of the feasible solutions to the problem until it reaches the best one. Solutions to integer programming problems (MIP/MIQCP) are usually represented as a tree and algorithms such as branch and bound navigate this tree efficiently, searching for the branch with the best solution. The initial (root) solution of a B&B algorithm is usually obtained by solving the simplex

algorithm on a relaxed version of the MIP where all discrete variables are relaxed and considered continuous.

A problem can be unfeasible, if no solution exists, or it can have a single global optimum solution if the best combination of decision variables can be identified. As a result of the analytical solving process, mathematical programming solvers for linear problems can indicate whether a formulated problem is unfeasible or how close a solution is to the global best. LPs can be solved in polynomial time while non-linear and integer programming are harder to solve and algorithms depend on the convexity of the objective function, size of the problem and the number of variables.

In this thesis, MINLP and MIP models are proposed for the problems of community detection in dynamic networks and for the development of QSAR models using regression and network analysis techniques. The MINLP of Chapter 3 is solved multiple times so as to efficiently select the best solution from multiple good approximations, while the MIPs in Chapters 4 and 5 are solved to optimality in all scenarios and therefore, do not require multiple runs.

2.2.1 Mathematical programming formulations of modularity metric

The modularity metric is an example of objective function for the optimisation problem of community detection. The formulation presented in Section 2.1.2 has been presented in the literature as an MILP [63], following the original description of the modularity metric. The decision variables of this MILP consisted of the binary variables $\delta_{i,j}$, indicating whether nodes i and j were in the same module. This formulation requires a post-processing step to decode the number of modules

identified and which nodes belong to which modules. MILP models can be solved to optimality but because of the complexity of this objective function, alternative models have been proposed to solve it.

Modularity can be represented as a MIQCP or as a MINLP [62, 64, 85, 86], directly expressing whether a node belongs to a module m. The decision variables can then be represented by the binary variables Y_{nm} which is set to 1 whenever a node n is assigned to module m and 0, otherwise. Modularity is then reformulated as [62]:

$$Q = \sum_{m} \left[\frac{L_m}{L} - \left(\frac{D_m}{2L} \right)^2 \right], \tag{2.7}$$

where $m \in M$ represents the modules, L_m is the number of edges or weighted sum of edges in the case of a weighted network for module m, L is the total number of links in the network or the sum of weighted links, and D_m indicates the sum of degrees, or strength, of nodes in module m.

This notation is more convenient because we can control the *maximum* number of modules allocated, reducing the computational complexity of the problem. The solver is still free to leave a module empty if less than M modules are necessary. A constraint of this model is that a node n can only be allocated to a single module:

$$\sum_{n} Y_{nm} = 1. \tag{2.8}$$

The remaining constraints define L_m and D_m:

$$L_m = \sum_{n} \sum_{\substack{e>n \\ e \in CN_n}} \beta_{ne} Y_{nm} Y_{em}, \tag{2.9}$$

$$D_m = \sum_n d_n Y_{nm}, \tag{2.10}$$

where CN_n is the set of nodes e connected to a node n, β_{ne} represents the weight between two nodes ($\beta_{ne} = 1$ in unweighted networks), and d_n is the degree of node n.

This mathematical model forms the basis of other MINLP and MIP models that have been explored and extended to detect disjoint [62, 64] and overlapping [87, 85, 88] communities as well as to communities in multilayer networks [25]. Other mathematical programming formulations for modularity and related metrics can be found in [63, 65, 66, 89–92].

2.2.2 Alternative metrics for community detection and mathematical programming formulations

Other metrics alternatives to modularity have been proposed in the community detection literature [93, 94]. This section describes three alternative examples of objective functions for which, similar to modularity, the partitions of a network are considered ideal when these metrics are minimised or maximised. Although these metrics are not used in the algorithms proposed in the next chapters of this thesis, these were used in a preliminary study presented in Section 2.2.3 to investigate the suitability of modularity.

Ratio Cut

The Ratio Cut was originally proposed in [95] to identify the optimal graph partitioning. Two types of links can be observed when clustering nodes in a

network: those linking nodes from the same community (intra-community) and those connecting nodes from different communities (inter-community, also called cut). The ratio cut is measures the proportion of inter-community links leaving a module m to the number of vertices in m [96]. The metric can be formulated in optimization notation as follows:

$$\begin{aligned} &\text{Minimize} && RC = \sum_{m} \frac{C_m}{(\sum_n Y_{nm})} \\ &\text{subject to} \\ &&& \sum_{m} Y_{nm} = 1, && \forall n \\ &&& C_m = \sum_{n,e} \sum_{\substack{m_0 \\ m_o \neq m}} Y_{nm} Y_{nm_o}, && \forall n, \end{aligned} \tag{2.11}$$

where Cut_m is the cut size of the module m to the rest of the network, i.e., the number of links to all nodes in other modules $m_o \mid m_o \neq m$. The denominator is simply the number of nodes n that belong to the module m.

Ratio Association

The ratio association is a simple metric representing the ratio of links connecting nodes inside the same module [97]. For community detection, it is desirable to find a partition that maximizes this ratio for each module [1].

A formulation of Ratio Association is given :

[1] Some authors define Ratio Association as function to be minimized, calling it Negative Ratio Association $NRA = -RA$.

$$\begin{aligned} &\text{Maximize} && RA = \sum_m \frac{L_m}{\sum_n Y_{nm} + \epsilon} \\ &\text{subject to} \\ &&& \sum_m Y_{nm} = 1, && \forall n \\ &&& L_m = \sum_n \alpha_n Y_{nm} + \sum_{\substack{n,e \\ n<e}} \beta_{ne} Y_{nm} Y_{em}, && \forall n. \end{aligned} \tag{2.12}$$

Community Score

Community Score was proposed in [98] and defines the community detection problem as that of finding sub-matrices in a graph such that the sum of densities of these sub-matrices is maximised. The most dense sub-graphs represent communities in a network, similarly to another metric, the modularity density [99]. Community score, however, measures density based on volume and row/column means of the adjacency matrix, incorporating more information about nodes interconnections [98].

A sub-matrix of the adjacency matrix A can be represented as $S = (I, J)$, in which I and J are subsets of rows and columns of A, respectively. The power mean $M(S)$ of S of order r is defined as:

$$M(S) = \frac{\sum_{i \in I} (A_{iJ})^r}{|I|}, \tag{2.13}$$

where A_{iJ} represents the mean value of the row i for the columns J in the matrix A; $|I|$ is the number of rows in the subset I and r is a parameter to the algorithm, typically 1.5.

The score Q of a sub-matrix S is defined as $Q(S) = M(S)vol(S)$ - where $vol(S)$ is the volume of the module $vol(S) = \sum_{ij} A_{ij}$. The community score metric for a partitioning of k sub-matrices is then defined as:

$$CS = \sum Q(S_i)_i^k \tag{2.14}$$

2.2.3 Comparison between community detection metrics

In a preliminary study to compare the suitability of community detection metrics, modularity and the three alternative metrics described above were implemented and tested on a set of benchmark networks.

A set of 8 synthetic networks with different community structures were generated using the LFR benchmark algorithm [100]. These benchmark networks were designed to test and validate community detection algorithms and have a main parameter, μ, which can be varied from $\mu = 0$ to $\mu = 0.50$. Networks generated with small μ are more community-like and have well-defined community structure while networks with $\mu = 1$ do not present distinguishable modular structure and connections are completely random.

In this small test, the following parameters were used to generate the networks:

Mixing Parameter (μ) From 0.10 to 0.50, in intervals of 0.05.

Size 500 nodes.

Average Degree 30.

Maximum Degree 100.

Minimum community size 13 nodes.

Maximum community size 100 nodes.

Exponent for the degree distribution 2.

Exponent for the community size distribution 1.

Each metric was then optimised to identify partitions for these networks. The partitions found were compared to the "ground truth" of the benchmark networks using with normalised mutual information (NMI), a measure that indicates how similar two partitions are. When $NMI = 0$, the two module assignments are not alike and when $NMI = 1$, the two partitions are identical.

Figure 2.4 shows that the modularity metric had the best performance among the tested metrics. Even in the cases where the community structure of the synthetic networks are not well defined, modularity still correctly assign the modules to almost all nodes in the network. Figures 2.5 and 2.6 illustrates the partitions identified for the synthetic network with $\mu = 0.40$ by modularity and the ratio association, respectively. Nodes are visually allocated close to other nodes in the same module. Note how the partition identified by ratio association includes nodes from different parts of the network which are not connected to each other, in the same module.

These preliminary investigations confirmed that, despite its limitations, modularity was still a more suitable and reliable method for community detection and it was selected as the default metric for the community detection techniques introduced in the rest of this thesis.

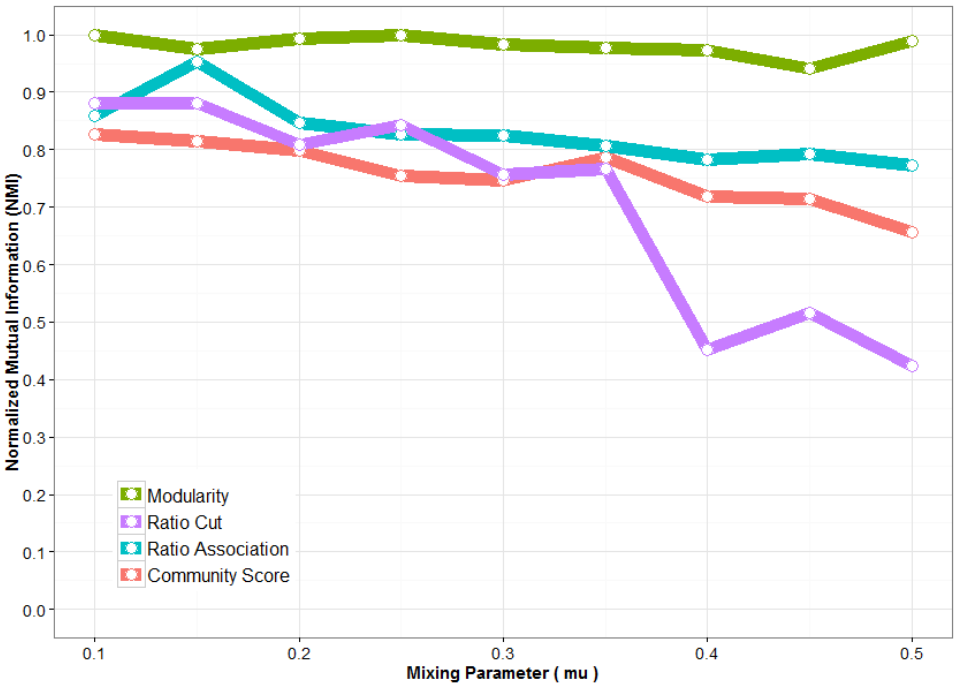


Fig. 2.4 Performance of community detection metrics for synthetic networks.

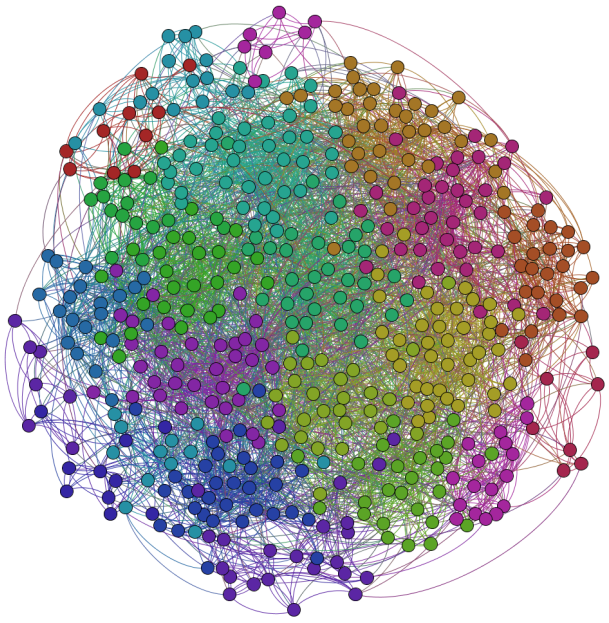


Fig. 2.5 Modules in LFR benchmark network ($\mu = 0.40$) identified by the modularity metric

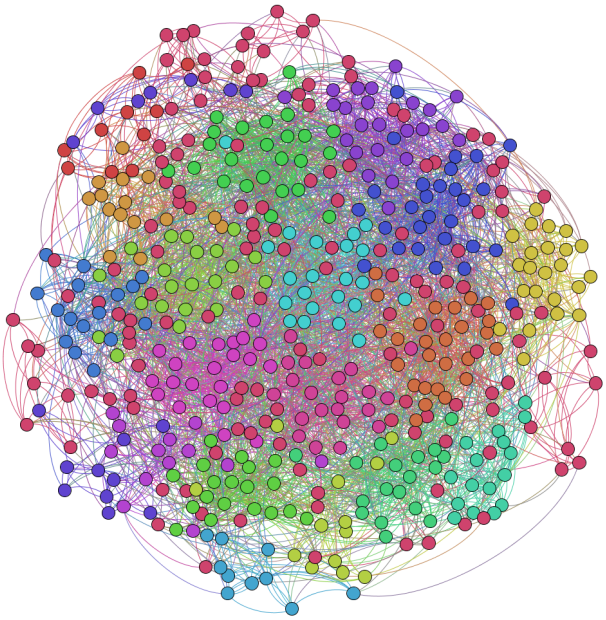


Fig. 2.6 Modules in LFR benchmark network ($\mu = 0.40$) identified by the ratio association metric

2.3 Quantitative Structure-Activity Relationship (QSAR)

Sections 2.1 and 2.2 have introduced some of the major concepts used throughout this thesis: properties of complex networks, in particular modularity, and principles of mathematical programming applied to community structure. The work described in Chapter 3 is directly linked to the concepts introduced in these sections. In the remaining chapters, methodologies were proposed not only for complex networks analysis but also for regression and machine learning.

One of the objectives of this thesis is to explore how optimisation techniques and network analysis could be combined to solve real problems in science in a transparent and flexible manner. One of the real applications, presented in Chapters 4 and 5, is in the field of drug discovery. More specifically, the algorithms presented therein tackle the problem of developing Quantitative Structure-Activity

Relationship (QSAR) models, regression algorithms used in medicinal chemistry, cheminformatics and bioinformatics to predict the potency of chemical compounds based only on the structural and topological properties of the molecules. This section gives an introduction to these models, the main elements, concepts and validation strategies relevant to this work and introduced in the next chapters.

2.3.1 Introduction to QSAR models

Designing a new drug is hard and expensive, and takes on average 10 to 15 years and nearly 1 billion US dollars to bring a new drug to the market [101]. The drug discovery process requires a multidisciplinary team to identify a biological target associated with a disease of interest and, from the vast and infinite chemical space [102], select a molecule that modulates the target so as to reduce or halt the biological processes of the disease. Computational and optimisation tools are widely employed in all these steps to explore research avenues and optimise drug candidates [103–107].

Quantitative Structure-Activity Relationship (QSAR) models are examples of these tools. These regression and classification models aim to predict biological activity of chemical compounds based on molecular structure [108]. QSAR is used early in the drug discovery process to screen for potential hits, to draw hypothesis from the data and to help identify the mechanisms of action of drug candidates [109]. But it can also be used later, at lead optimisation stage. This is the process of making minor modifications in promising compounds to improve its pharmacokinetic properties: absorption, distribution, metabolism, excretion and toxicity in the body (ADMET) [110–115]. QSAR can even used to help re-purpose existing medicines to different treatments [116].

The history of QSAR models can be traced back to the work of Corwin Hansch and his collaborators in the 1960s [117–120]. These publications showed that biological activity of chemical compounds could be expressed as a mathematical function of physical and chemical properties of these molecules [121], under the assumption that small modifications in the structure of molecules lead to a proportional change in biological activity. These models were built for small series of similar compounds using linear regression with few quantitative features and aimed to discover a transparent relationship between molecular structure and biological activity [122].

This approach is still employed successfully to design new drugs [123, 124] but most recent models consist of hundreds or thousands of molecular descriptors calculated from the chemical, 2D or 3D representations of the molecules [125–128] and are often built with non-linear algorithms such as neural networks, support vector machines with Gaussian kernels and random forest [129]. In these situations, interpreting QSAR models is as difficult as the interpretation of these "black-box" machine learning algorithms [130–132]. The emphasis of these models is generally on obtaining better accuracy of prediction, not understanding how the chemical structure is associated with biological activity [133]. Some algorithms allow the creation of a ranking identifying the importance of the descriptors used in the QSAR model, as is the case of random forest out-of-bag estimation. Similarly, other methods have been proposed in the literature to post-process the results of "black-box" algorithms in order to interpret their results [132, 134–136]. However, this post-processing step adds yet another layer of complexity to the interpretation of the models. There is therefore potential to explore the creation of QSAR models that are interpretable and transparent at the same time. The creation of models that follow these principles is one of the objectives of this thesis.

2.3.2 Molecular Descriptors

A chemical compound can be described and identified in various ways. Acetaminophen, or paracetamol, for example could be described by:

- its molecular formula: $C_8H_9NO_2$ or $HOC_6H_4NHCOCH_3$,
- the SMILES code: CC(=O)NC1=CC=C(C=C1)O,
- the International Chemical Identifier: InChI=1S/C8H9NO2/c1-6(10)9-7-2-4-8(11)5-3-7/h2-5,11H,1H3,(H,9,10),
- or by its 2D structure, shown as a graph in Figure 2.7.

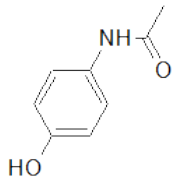


Fig. 2.7 2D Structure of acetaminophen

Similarly, there are a number of ways to describe properties of molecules. In QSAR, attributes of chemical compounds are called molecular descriptors and consist of numerical values that capture aspects of the chemical structure to be correlated with biological activity [101]. Hundreds and even thousands of descriptors can be calculated using available open source and commercial software such as PaDEL [137], CDK [138], RDKit [139], Dragon [140] and MOE [141].

Molecular descriptors can represent fragments (number of rings of a specific size, predefined substructures), the connectivity between types of atoms (e.g. average distance between primary carbons), structural (molecular weight, number of rotatable bonds), topological properties extracted from the molecular graph (eigenvalues, average shortest distances) [142] or properties related to the 3D

structure of the compound. Table 2.1 shows example of molecular descriptors calculated for acetaminophen using the Chemistry Development Kit (CDK) library.

Count descriptors				*Continuous descriptors*			
nSmallRings	*nAromRings*	*nAtomLC*	*khs.tCH*	*Zagreb*	*WPATH*	*MDEO.11*	*BCUTw.1l*
1	1	8	0	50	166	0.75	11.9963

Table 2.1 Examples of molecular descriptors calculated for acetaminophen

Some descriptors are easier to interpret than others. From the example illustrated in Table 2.1, count descriptors could be easily inspected in Figure 2.7. *nSmallRings* represent the number of small rings from size 3 to 9, *nAromRings* indicates the number of aromatic rings, the number of atoms in the largest chain is given by descriptor *nAtomLC* and *khs.tCH* represents the number of occurrences of a Kier-Hall fragment [143, 144]. In particular, *khs.tCH* represents a carbon with a triple bond to another atom and since no such bond exist in the molecule, the descriptor value is zero. Topological indices and descriptors that encode properties of the molecular graph or averages are usually harder to interpret. In the example above, the *Zagreb* index is given by the sum of the squares of atom degrees $Zagreb = \sum_i \delta_i^2$, *WPATH* is defined as half the sum of distances $\mathbf{D}$ between all pairs of nodes (i, j) in the molecular matrix: $WPATH = \frac{1}{2}\sum_{ij} \mathbf{D}_{ij}$, *MDEO.11* is a metric related to the average distance between primary oxygen atoms and *BCUTw.1l* is one of the BCUT descriptors defined as one eigenvalue of a modified version of the molecular graph. Despite the interpretability limitations, these descriptors are easy to compute and were shown to correlate with many biological properties such as water solubility, molecular shape, boiling point and partition coefficient [145].

The descriptors presented above are just a few examples of the thousands of attributes that can be used to construct a QSAR model. Note, however, that including too many attributes can complicate the analysis and interpretation of

models. In some cases, when conducting a small study of congeneric molecules where certain fragments are believed to have an impact on biological activity, one can rationally select a group of appropriate attributes to create a QSAR model. Whereas, when one wants to model a large set of heterogeneous molecules or when hypotheses about the structure-activity relationship do not exist yet, the selection of a subset containing the most relevant features is a hard combinatorial problem that can not always be done manually. In these cases, automated feature selection techniques are particularly useful and can be performed in two ways: as a filter before the QSAR model is created, or as a wrapper embedded within the regression or classification algorithm used to create the model [146]. In the algorithms proposed in Chapters 4 and 5, the wrapper method is used to select features within the algorithm simultaneously with the creation of the transparent models that make up the QSAR model.

2.3.3 Validation procedures

QSAR models have an impact on decisions about human health and the environment. Therefore, these models must undergo robust validation procedures so as to eliminate spurious correlations and chance findings [147, 148]. For regulatory purposes, the Organization for Economic Cooperation and Development (OECD) have proposed the following principles every QSAR models should follow:

A defined endpoint: QSAR models should have a consistent metric of the biological activity studied and should take into account the variability when data comes from different laboratories.

An unambiguous algorithm: Descriptors should be comprehensible and must not be collinear to other descriptors in the same QSAR model. The algorithm should be reproducible.

A defined domain of applicability: Predictions made by a QSAR model are only applicable to molecules related to those used to train the model. There should not be duplicated entries in the data set and the range of biological activity should not be too narrow.

Appropriate measures of goodness-of-fit, robustness and predictivity: Data set should be split into training and test sets and validated accordingly. Model should not overfit trained data.

A mechanistic interpretation, if possible: QSAR models should potentially be able to explain the biological activity of compounds.

To perform the validation described above, QSAR data sets are divided into two main parts: internal set, used to develop the QSAR model, and external sets, to validate the capacity of the QSAR model to generalise to unseen data. The model is usually trained using k-fold cross-validation (CV), where the internal set is divided into folds, multiple instances of training and test (or validation) sets, the training samples are used to train the algorithm and its accuracy is assessed by samples in the internal test set. Typically, k=5 or k=10 folds are used, resulting in folds where 80% and 90% of the internal data is used to create a QSAR model, respectively, while the remaining is in the test set [108].

2.3.4 Molecular Similarity

The similarity property principle states that similar small molecules must equally have a similar biological activity. This principle permeates computational chem-

istry fields and is one of the main assumptions of QSAR. Molecular similarity is used to perform virtual screening in databases, to searching for similar compounds or even to identify compounds with a different core structure but similar activity [149]. To assess similarity, molecules should first be represented in a convenient numeric format to allow for the comparison of attributes so that if two molecules share a large number of common features, these are considered similar. Even though molecular descriptors such as the ones cited in Section 2.3.2 are suitable for the task, in most applications, molecular fingerprints are the preferred representation.

Molecular fingerprints are binary arrays of a fixed size that characterise a molecule. Two of the most popular fingerprinting techniques are MACCS keys and the extended-connectivity fingerprints (ECFP). MACCS keys fingerprint technique is a type of substructure or dictionary-based fingerprinting technique and consist of 166 bits. Each bit represent a fragment that can be found in a molecule, ON and OFF bits indicate the presence or absence of a fragment in the molecule, respectively. By contrast, the extended-connectivity fingerprints (ECFP) technique identifies each atom with an integer value; then, it iteratively identifies adjacent atoms, starting from the immediate neighbourhood (radius 1) to a maximum radius R. The result is a list of integer values that can be converted to a binary vector of a predetermined size using a hashing function [150]. The default radius used is 4 (ECPF4) and the binary array is usually fixed at a length of 1024 bits.

Bits on ECFP fingerprints are not directly interpretable as dictionary-based fingerprints and there is a risk of bit collision, when the same binary sequence represents two different substructures. But despite those limitations, ECFPs are more advantageous for SAR studies than dictionary-based fingerprints. The predefined substructures in these fingerprints were designed for substructure

searching, have a limited scope and might not contain fragments related to novel chemical variation relevant to a library of compounds.

Regardless of the fingerprint technique used, it is easy to see that similar molecules share a large number of bits in common. The Tanimoto coefficient (Tc) or Tanimoto similarity or Jaccard coefficient is a metric of similarity between two fingerprints F_A and F_B that represents this intuitive notion and is given by:

$$Tc = \frac{|A \cup B|}{|A| + |B| - |A \cup B|}, \tag{2.15}$$

where $|A|$ is the number of ON bits in F_A, $|B|$ is the number of ON bits in F_b and $A \cup B$ represents the common ON bits in both fingerprints. When the two fingerprints are identical, the Tanimoto coefficient is maximum $Tc = 1$, when there are no common ON bits between the molecules, $Tc = 0$. Other similar similarity metrics are Dice index, Cosine coefficient and Sorgel distance [151].

2.3.5 Activity cliffs

One of the main assumptions in QSAR is that structurally similar molecules have a similar biological activity. However, that is not always the case. A small change in the structure of a compound can produce a large variation in potency, a discontinuity known as activity cliff (AC) [152]. The change of chirality or position of a single atom, the modification of a functional group or other subtle changes in the scaffold of a compound can alter the potency of a compound 10 or 100-fold [153–156].

Activity cliffs can be explored using activity landscape visualisation, using quantitative measures or using networks. In network representation of ACs,

molecules are linked according to their similarity using fingerprints and the Tanimoto coefficient [157]. Only links above a predefined threshold $Tc = t_\alpha$ are considered and a discontinuity score is calculated for each node using the following formula:

$$\text{raw}_{\text{disc}}(i) = \frac{\sum_{\{j|\text{Tc}(i,j)>t_\alpha, i\neq j\}} \text{potdiff}(i,j) \times \text{Tc}(i,j)}{|\{j|\text{Tc}(i,j) > t_\alpha, i \neq j\}|} \tag{2.16}$$

where $\text{potdiff}(i,j)$ is the absolute potency difference between compounds i and j and $\text{Tc}(i,j)$ the Tanimoto similarity of fingerprints calculated for these compounds. Raw scores are then converted to Z-scores and normalised to [0,1] range using cumulate probability distribution of a normal distribution. Samples can then be assigned to a category of activity cliff according to the discontinuity score [158]: low ($disc \in [0.0, 0.4)$), intermediate ($disc \in [0.4, 0.7)$) and high ($disc \geq 0.7$).

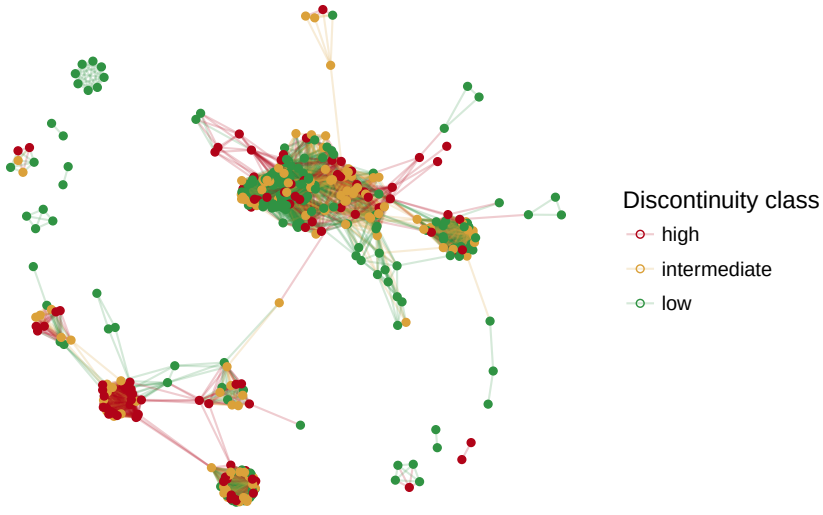


Fig. 2.8 Example of molecular network where nodes represent molecules and links indicate that molecules are similar above a predefined threshold. Each node is coloured according to the activity cliffs in its neighbourhood.

An example of a network representation of activity cliffs can be seen in Figure 2.8. The network expresses the similarity between a group of inhibitors

of muscarinic 3, an acetylcholine receptor found in the brain and neuromuscular junctions. The visualisation helps to identify clusters and neighbourhoods in the network where intermediate (yellow) and high (red) activity cliffs occur. In this thesis, the modules of these networks are also explored as a tool for QSAR modelling, as described in Chapter 5.

2.4 Summary

This chapter introduces three main concepts related to the work described in the following chapters: mathematical programming, network analysis and the development of quantitative structure-activity relationship (QSAR) models. Three methods are proposed to identify the evolution of communities in temporal networks, predict biological activity from the separation of chemical compounds into groups and the exploration of network properties to create QSAR sub-models. All algorithms are represented as mathematical programming at some level.

Chapter 3

A Mathematical Programming Approach for Sequential Clustering of Dynamic Networks

Foreword

The modularity metric is one of the most used metrics by community detection algorithms in complex networks and it has been successfully represented as mixed integer programming models to detect communities in static networks, as described in Section 2.2.1. This chapter introduces a new optimisation model based on modularity to detect groups in dynamic networks, where the connections between nodes change over time.

The proposed algorithm optimises two objective functions simultaneously: it maximises the modularity of each snapshot of the network while also considering the modular structure of previous time steps. The rationale is that modules are not

isolated patterns of a network but are rather a reflection of the previous interactions in the network. By considering the history of connections, the algorithm is capable of identifying more permanent clustering patterns than if each time step was considered individually or aggregated into a single network slice.

The content of this chapter has been published as:

Jonathan C. Silva, Laura Bennett, Lazaros G. Papageorgiou, and Sophia Tsoka. A mathematical programming approach for sequential clustering of dynamic networks. The European Physical Journal B, 89(2):39, feb 2016. DOI: 10.1140/epjb/e2015-60656-5.

JCS was the main author of the paper and was responsible for data collection, implementation and analysis of data. The idea for this paper was conceived and analysed in collaboration with LGP, LB and ST.

Abstract

A common analysis performed on dynamic networks is community structure detection, a challenging problem that aims to track the temporal evolution of network modules. An emerging area in this field is *evolutionary clustering*, where the community structure of a network snapshot is identified by taking into account both its current state as well as previous time points. Based on this concept, we have developed a mixed integer non-linear programming (MINLP) model, SeqMod, that sequentially clusters each snapshot of a dynamic network. The modularity metric is used to determine the quality of community structure of the current snapshot and the historical cost is accounted for by optimising the number of node pairs co-clustered at the previous time point that remain so in the current

snapshot partition. Our method is tested on social networks of interactions among high school students, college students and members of the Brazilian Congress. We show that, for an adequate parameter setting, our algorithm detects the classes that these students belong more accurately than partitioning each time step individually or by partitioning the aggregated snapshots. Our method also detects drastic discontinuities in interaction patterns across network snapshots. Finally, we present comparative results with similar community detection methods for time-dependent networks from the literature. Overall, we illustrate the applicability of mathematical programming as a flexible, adaptable and systematic approach for these community detection problems.

3.1 Introduction

Complex networks exhibit several topological features that distinguish them from more simple networks such as lattices and random networks. One feature in particular is the tendency of nodes to organise themselves into a modular topology, known as community structure [159]. The detection of such communities, also known as modules, is widely accepted as means of revealing the relationship between topological and functional features of complex systems [61].

Network representations of complex systems are often static, corresponding to either a snapshot of a system at a certain point in time or an aggregation of data over multiple time points. However, in reality networks are not static; nodes and interactions can be created or cease to exist. For example, in social networks friendships are made and broken. In a business, employees retire and new members of staff are employed. In biological systems, not all interactions

take place at the same time, depending upon spatial, temporal or environmental conditions [160].

A dynamic network is defined as a series of network snapshots at two or more time points, where time can represent seconds, days, years or various states of a system. The changes which occur at the node and interaction level may affect global measures and descriptions of the network e.g. community structure as modules are created, destroyed, split into multiple groups or merged together. Consequently, incorporating temporal information into network modelling frameworks may lead to more accurate representations of complex systems. It follows that a current challenge in community structure detection is the identification of modules in dynamic networks.

Community structure in relation to dynamic networks has been tackled in various ways. Consensus clustering attempts to find a partition of a system that is to some extent relevant at each time step [161, 162, 25]. Alternatively, many approaches cluster the static snapshot networks independently and employ various methods of comparison between partitions to quantify change in community structure or follow the evolution of communities [21, 163–165]. In particular, some methods aim to detect drastic discontinuities in community structure which represent some form of important 'event' [166, 167].

Where snapshots are clustered individually, historic community structure is not taken into account. It has been proposed, however, that the community structure of the network at time t should not be taken as independent of the community structure of the network at time $t-1$. In other words, a network at time t is clustered with respect to a known partition of the network at $t-1$. Such methods take advantage of information about the community structure of

previous snapshots in order to infer the structure in the current time step, as expressed by evolutionary clustering approaches.

The first evolutionary clustering method introduced the idea of temporal smoothness, where the snapshot quality measure (e.g. modularity) is maximised at the current time point and a distance measure, the history cost (e.g. mutual information, rand index etc.) between the current snapshot and the previous snapshot is minimised [168]. A trade-off is therefore made between remaining faithful to the current data, but minimising the variation between the current partition and the previous one. Similar approaches can be seen in [169–176]. In this study, we present a mathematical model that inherits this framework and our analysis focuses on data that is sequential in nature.

Mathematical programming provides a flexible and intuitive modelling framework that has been shown to be competitive in numerous community detection algorithms [62–66, 85, 59, 88, 25]. In our previous work we have proposed mathematical programming methods based on modularity optimisation to identify hard partitions [62, 64, 85], overlapping communities [88] and to cluster multiplex networks [25]. Here, we extend our previous work and propose a mathematical programming approach to evolutionary clustering. In particular, we report a mixed integer non-linear programming (MINLP) model that given a series of network snapshots, returns a partition for each of the snapshots, taking into account historical information. The applicability of our method is demonstrated through its application to synthetic and real networks and through a comparative analysis with similar methods from the literature.

3.2 Methods

3.2.1 A mathematical programming model for sequentially clustering snapshots of dynamic networks

Here we report an MINLP model of evolutionary clustering that sequentially identifies the community structure of each snapshot in a dynamic network. The indices, parameters and variables associated with our model, known as SeqMod, are given below:

Indices

n, e nodes (the union from all input snapshots)

m modules

t time step

Parameters

β_{net} weight of link between n and e at time t

d_{nt} weighted degree (strength) of node n at time t

L_t sum of the weights in the network at time t

$\gamma_{ne,t}$ equals to 1 if nodes n and e are in the same community at time t

ε the smoothness control, a user parameter that indicates the weights given to the preservation coefficient and modularity

Binary Variables

Y_{nmt} equal to 1 if node n is in module m at time t; 0 otherwise

Continuous Variables

D_{mt} the sum of the weighted degree of nodes in module m at time t

L_{mt} the sum of the weights of links that are in module m at time t

Modularity expresses how well-defined the community structure of a network is [43]. The metric provides an intuitive description of community structure and is one of the most popular methods for community structure detection. We thus employ modularity to determine the community structure of the current snapshot:

$$Q_t = \sum_m \left(\frac{L_{mt}}{L_t} - \left(\frac{D_{mt}}{2L_t} \right)^2 \right), \quad \forall t, \tag{3.1}$$

where L_{mt} is the sum of weights of links within module m at time t:

$$L_{mt} = \sum_{\substack{n,e \\ n>e}} \beta_{net} Y_{nmt} Y_{emt} \quad \forall m, t, \tag{3.2}$$

and D_{mt} is the sum of weighted degrees of nodes in module m at time t:

$$D_{mt} = \sum_n d_{nt} Y_{nmt} \quad \forall m, t. \tag{3.3}$$

The central idea in Evolutionary Clustering [168] is to detect community structure that is consistent with current data and tracks changes smoothly over time, i.e., the community structure does not change drastically from a time step to

another. This usually means that the community structure at a specific time step t should reflect the data at t as well as the community structure at the immediate previous time step. Here, however, we take a different reference time step t^* that has had the maximum modularity up to the current time step and we define a preservation coefficient, Δ_t, to measure the number of pairs of nodes that are co-clustered both at t^* and at the current time step t:

$$\Delta_t = \sum_m \sum_{\substack{n,e \\ n>e \\ \gamma_{ne,t^*}=1}} Y_{nmt} Y_{emt} \quad \forall t. \tag{3.4}$$

This preservation coefficient represents how similar two partitions are to each other. Δ_t and Q_t can be competing objectives and we define the parameter ε for controlling the influence each metric has over the optimization process. This parameter is said to control the smoothness of the transitions, i.e. the influence of the historical clustering information has over the current network structure. The objective function is defined according to the definition of evolutionary clustering in [169] :

$$(1-\varepsilon)Q_t + \varepsilon\Delta_t \quad \forall t. \tag{3.5}$$

Parameter ε is restricted to the interval $[0, 1]$, such that when $\varepsilon = 0$, the partition at t^* does not influence the clustering of the partition at the current snapshot, at t. If $\varepsilon = 1$, our model would simply maintain the previous partition and the modularity of the current snapshot would not be considered. This provides for a more intuitive setting of the expected smoothness of transitions.

Both Q_t and Δ_t are therefore normalised. Q_t ranges from $-\frac{1}{2}$ to 1 [56] and Δ_t ranges from 0 (when no pairs of nodes were maintained from the previous time

step to the current one) to $\Delta_{t_{max}} = \sum \gamma_{ne,t^*}$, when all pairs of co-clustered nodes from the reference time step remain together on a module at the current time step. Our objective function is therefore redefined as:

$$\frac{(1-\varepsilon)}{3}(2Q_t + 1) + \frac{\varepsilon}{\sum \gamma_{ne,t^*}}\Delta_t \tag{3.6}$$

Finally, SeqMod detects disjoint communities in each snapshot of the network, i.e. each node belongs to only one module at time t. Therefore we add the following constraint:

$$\sum_m Y_{nmt} = 1 \quad \forall n, t. \tag{3.7}$$

The complete model is formulated below:

maximize	$\frac{(1-\varepsilon)}{3}(2Q_t + 1) + \frac{\varepsilon}{\sum \gamma_{ne,t^*}}\Delta_t$	
subject to		
constraints	(3.1, 3.2, 3.3, 3.4, 3.7)	
	$L_{mt}, D_{mt} >= 0$	$\forall m, t,$
	$Y_{nmt} \in \{0, 1\}$	$\forall n, m, t,$

SeqMod is implemented in GAMS (General Algebraic Modelling System) [76] using standard branch and bound (SBB) method as the mixed integer optimisation solver and CONOPT as the NLP solver with default parameters. To ensure that we give a reasonable representation of solution space, for each clustering experiment

(each with a different ε), the MINLP is solved iteratively 100 times, each time with a different random initial partition. After each time step is solved, the solution with the largest value of the objective function is selected. Even though an upper bound for the number of modules is provided, it is stressed that the actual number of modules in the partition is decided by the model.

3.2.2 Network data

SeqMod is tested on synthetic and real networks, summarised in Table 3.1. First, as an illustrative example, we have adapted a small synthetic network from [177]. This network comprises 22 nodes and 3 time steps. The network has 3 clearly defined communities at $t = 1$ but connections between two of them become increasingly denser at $t = 2$ and $t = 3$.

Network	Nodes	Time steps	Classes
Synthetic network	22	3	3
High School	180	7	5
MIT Social Evolution	61	36	8
Brazilian Congress	589	12	2

Table 3.1 Summary of networks used in this paper.

We have also tested real social networks, generated from proximity data among high school students, college students and members of the Brazilian Congress. The High School network represents the interactions among 180 high school students, over a period of 7 school days [178]. Students had to wear a device that would record any face to face contact with another student that lasted at least 20 seconds. The results were used to generate daily networks. The authors of the study have shown that about 91.5% of contacts made between students in this high school during the study involved students in the same class.

The third network was built from the MIT Social Evolution dataset [179], consisting of records of eighty MIT students who lived in the university dormitory during an academic year. Various types of interaction data (WiFi location, Bluetooth proximity, SMS and Calls exchanged) were collected using mobile phones that were given to the students. We selected Bluetooth proximity data to generate a dynamic network of weekly snapshots for 36 weeks. We restricted our network to the students who had their dormitory sector and year of studies mapped by the researchers and used only the records that had a probability of at least 20% that the involved students lived on the same dormitory floor. The final network, after applying these restrictions, had 61 nodes.

The fourth network was built from a dataset of the records of the roll call votes in the Brazilian Chamber of Deputies (the lower house of the Congress)[1]. To construct the networks, we first selected records in the period from 2003, the first year of the current ruling party government (PT) to 2014. Duplicated entries of each congressman were removed, results were ranked according to parties and records of the topmost four parties were selected. Two of these parties are in the opposition group (PSDB and DEM) while the other two are allied to the government (PT and PMDB). For every year in the records, we computed a weighted adjacency matrix based on [180] in which each cell represents the similarity of the votes of each pair of congressmen during that year. Unanimous vote sessions, i.e. cases where 95% of the present congressmen vote the same, were removed.

[1]The data was downloaded from https://github.com/estadaodados/basometro

3.2.3 Alternative clustering methods

We compare our results with other algorithms for community detection through evolutionary clustering, namely estrangement confinement, FacetNet and DynMOGA. The estrangement confinement algorithm [174] solves the snapshots sequentially using modularity and a measure of dissimilarity between two partitions called estrangement, which is proportional to the number of intra-community edges that becomes inter community over time. FacetNet [171] is based on a stochastic block model for the detection of communities and a probablistic model based on Dirichlet distribution that detects the evolution of the communities. DynMOGA is a multiobjective genetic algorithm that optimizes both modularity and normalized mutual information (with respect to the previous time step) [175].

We also compare our method to genLouvain [181], a robust algorithm with a modified version of modularity designed for multilayer networks. In order to obtain results that are comparable to an evolutionary clustering framework, we apply genLouvain sequentially at each time step, i.e. to detect the partitions for the network at t1, we input the first snapshot; for the second time step, we provide t1 and t2 as input, and follow a similar procedure for all other ensuing time snapshots.

3.2.4 Adjusted Rand Index

In order to evaluate the accuracy of SeqMod in detecting the ground truth community structure, we employ the Adjusted Rand Index (ARI) metric [182], a measure of similarity between two partitions, comparing it to a null probabilistic model. ARI yields 1 for identical partitions and usually 0 for independent partitions,

although it may also result in some negative values [183]. We used the mclust package implementation of ARI in R [184].

3.3 Results and discussion

3.3.1 Synthetic network

For illustrative purposes, we tested our model on a small synthetic network, shown in Figure 3.1. We note three clear distinct communities (one on the left and the other two on the top, and bottom right, respectively). The two rightmost communities have increasingly denser connections between them at t_2 and t_3. We investigate whether SeqMod can maintain the three communities over time and under what values of ε.

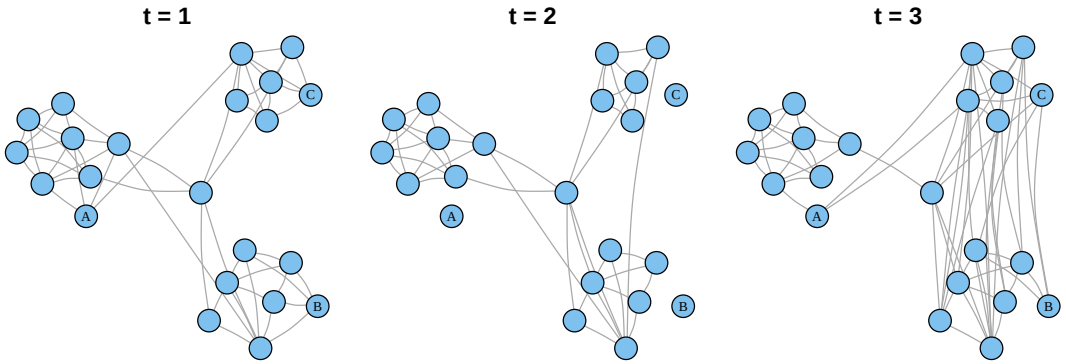


Fig. 3.1 Synthetic Network

If we fix $\varepsilon = 0$ for all time steps, the model does not consider historic information and each time step is solved sequentially considering only modularity. The result is shown in Figure 3.2. At t_1, three communities are detected, at t_2 the network is just slightly different, now the central node is placed in the red community instead of the blue. This reflects the minor changes in this snapshot, this node changed membership because at this time step it has more connections

with the red community than with the others. At t_3, there are now more edges between the red and blue communities and the community structure is different from the previous time steps. The three nodes marked A, B and C on the plot either did not interact with any other node at t_2 or are not present in the network at that time step, in our model we treat such nodes as isolated nodes. Since no historic information was taken into account, these nodes are assigned each to a singleton community at that time step.

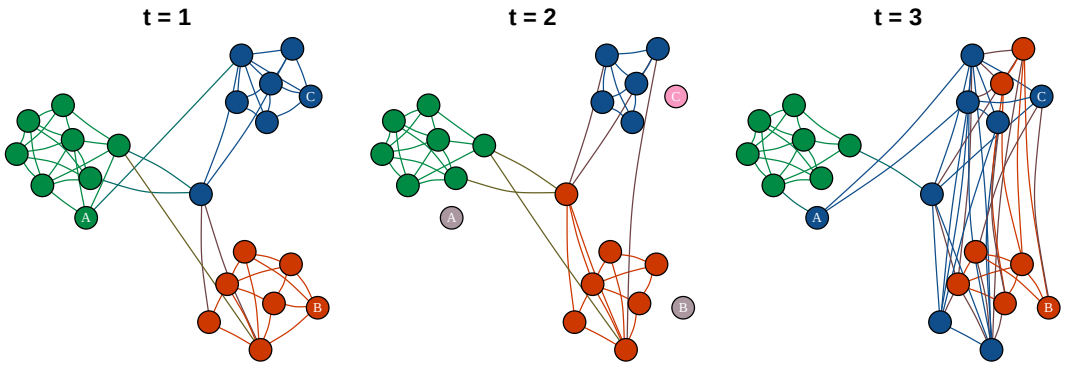


Fig. 3.2 Partitions detected by SeqMod for the synthetic network, using $\varepsilon = 0.00$.

The results are different when $\varepsilon > 0$, Figure 3.3 shows the partitions detected by SeqMod for the synthetic network for $\varepsilon \geq 0.10$. Community structure is now maintained from t_1 to t_2 even though the network has changed and at t_3, SeqMod detects a merge between the red and blue communities. Note also that the absent nodes at t_2 are maintained in their original community (from t_1). This is an advantage of the evolutionary clustering approach: it takes advantage of prior knowledge to infer the node community allocation even in the snapshot where it is not present/interacting.

Node A is a good example of a "promiscuous" node and its community membership is determined by the parameter ε. For example, if we fix $\varepsilon = 0.015$ the node is assigned to the green community at t_1 and is kept green at t_2, when it is absent, but on the third time step it belongs to the blue community since

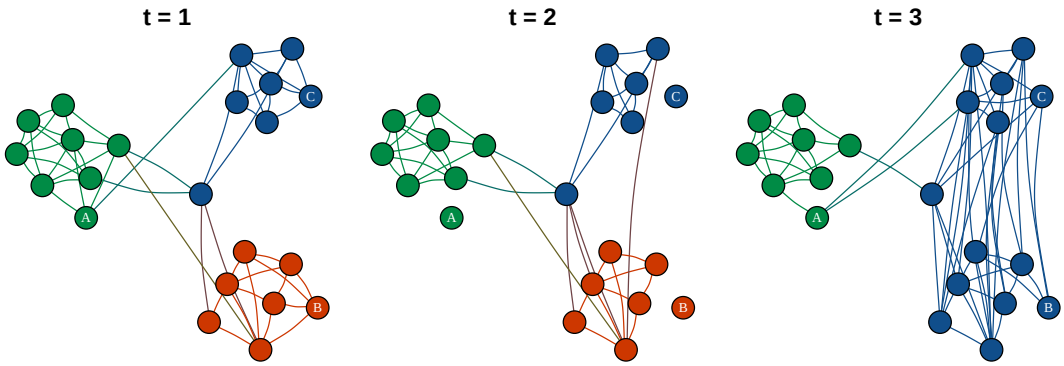


Fig. 3.3 Partitions detected by SeqMod for the synthetic network, using $\varepsilon \geq 0.10$.

now it has more ties with that group. The desired smoothness is controlled by the user parameter.

3.3.2 High School network

Our first real network is a proximity network created by aggregating information on face-to-face contacts between high school students. Students belong to one of five classes, each with a specific discipline programme: two **MP** classes, with a focus on mathematics and physics; two **PC** classes, with a focus on physics and chemistry and one **PSI** class, with focus on engineering. We take these classes as our ground truth communities and we investigate how closely our model detects them. Students are more likely to interact with other members of the same class [178].

We tested the High School network with a maximum number of communities set to 10 and Figure 3.4 shows how the difference to the ground truth network (expressed by ARI) changes over time according to the value of ε. We tested ε in the range $[0.00, 0.50]$, in intervals of 0.05. At time t_1, since there is no previous historic information, all test cases found the same partition but for the next time

steps, all test cases for $\varepsilon \leq 0.30$ have always improved ARI over the partitions with $\varepsilon = 0.00$.

This means that by properly setting the user parameter, it is possible to gain more information about the true communities of a network than with a static method. Furthermore, our dynamic community detection method is more accurate than applying a static community detection method to the aggregated network for this example; the Louvain algorithm [57] finds a solution with ARI = 0.763, less than the average result for SeqMod using $\varepsilon = 0.15$ (ARI = 0.816).

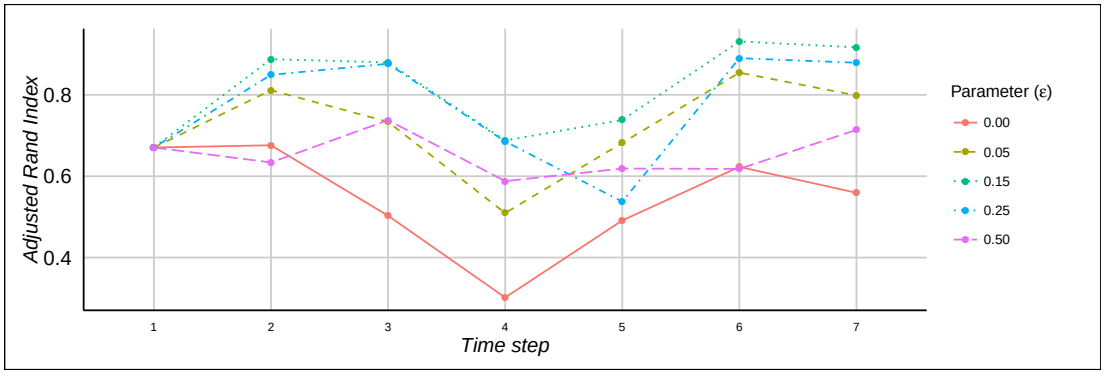


Fig. 3.4 High School network - The change in Adjusted Rand Index (ARI) at each time step for SeqMod depending on the value of ε.

Figure 3.5 shows the evolution of the network community structure for $\varepsilon = 0.15$. Notice that during t_4 and t_5 the red and blue communities are merged, they correspond to the two **MP** classes in the network, showing that students are also more likely to interact with others in similar disciplines than with the rest as previously noted by [178]. In all test cases where $\varepsilon > 0.05$ and $\varepsilon < 0.30$, this pattern occurs to some extent at these time steps; it can also be perceived from Figure 3.5 where for these values of ε, there is drastic change in interaction patterns at t_4 and t_5 (drop in ARI) that is followed by an increase in the metric for the next time steps (t_6 and t_7). This seems to suggest that these classes were more involved in activities together during these days or even that these students

are more likely to interact on Thursdays (t_3) and Fridays (t_4) than on Mondays (t_1, t_6), Tuesdays (t_2,t_7) and Wednesday (t_3).

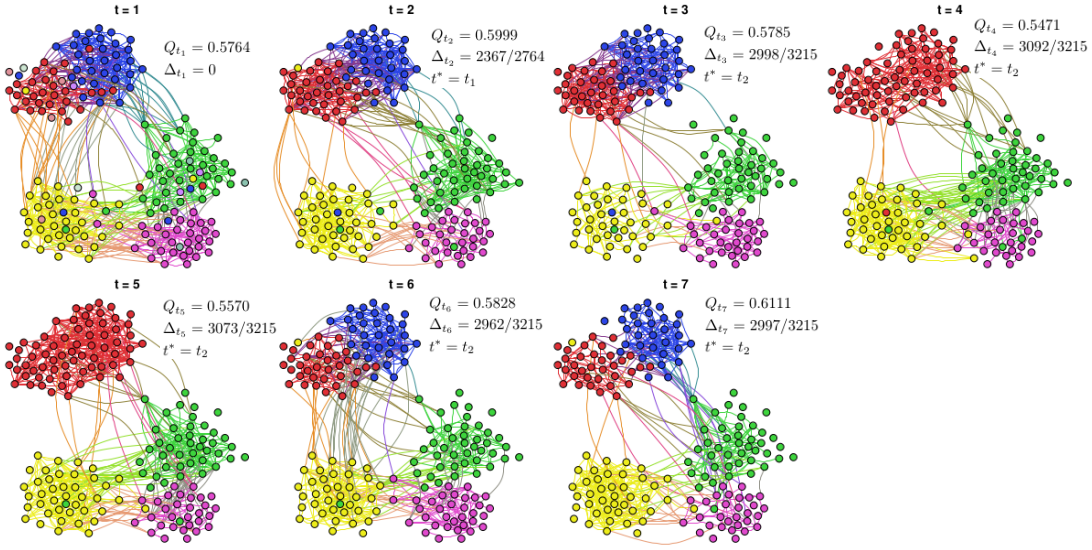


Fig. 3.5 Partitions detected for High School network by SeqMod, with $\varepsilon = 0.15$.

3.3.3 MIT Social Evolution dataset

The network built from Bluetooth proximity records from the MIT Social Evolution dataset is smaller than the previous network in terms of nodes but represents a longer time period. The interactions of the students in the Social Evolution network are registered based on Bluetooth proximity data rather than face-to-face contacts, as in the High School network, so some of the records may actually correspond to "false interactions". For example, if there is a record representing an interaction between students A and B, it may mean that they were in fact in different floors or room and not interacting at all. Therefore we restricted our network to those interactions with at least 20% of chance that they were in the same floor.

The dormitory sector where each student lived was reported to be the primary factor in defining the relationships between students [179], so we take the dormitory sectors as the ground truth communities for this network. Figure 3.6 shows the ARI results for various values of ε.

Once again, by incorporating historic information (any $0 < \varepsilon < 0.20$) the model yields results closer to the ground truth than clustering each snapshot individually. For $\varepsilon = 0.05$, the model improves in ARI over the results for $\varepsilon = 0.00$ and it still adapts to some major changes in community structure in the network. There is a pattern of decrease in ARI during the intervals $\{t_{12} - t_{16}, t_{19} - t_{21}, t_{26} - t_{28}\}$, the first of them correspond to the weeks of Christmas and New Year's Eve Holidays and the network is considerably reduced during these time steps (about 80% of students were absent). This explains why the model is more capable of matching the ground truth for relatively larger ε during the Holiday season: since the majority of the network consists of absent nodes, the community structure found previously is maintained.

We also note the trade-off nature of the smoothness control in the model. For this network, when $\varepsilon = 0.10$ the model yields the best result in terms of matching the true classes of the students (mean ARI = 0.371) but as it can be noticed in Figure 3.6, when solving with $\varepsilon = 0.05$ the model is better at detecting the changes in the network, ARI decreases during the holiday periods. One has to balance between the current state of the network, incorporating "ground truth" information sequentially, while also allowing for change. If ε is set too large, for example 0.50, the model might incur "over-smoothing", i.e. being unable to adapt to changes in the community structure on this period [185].

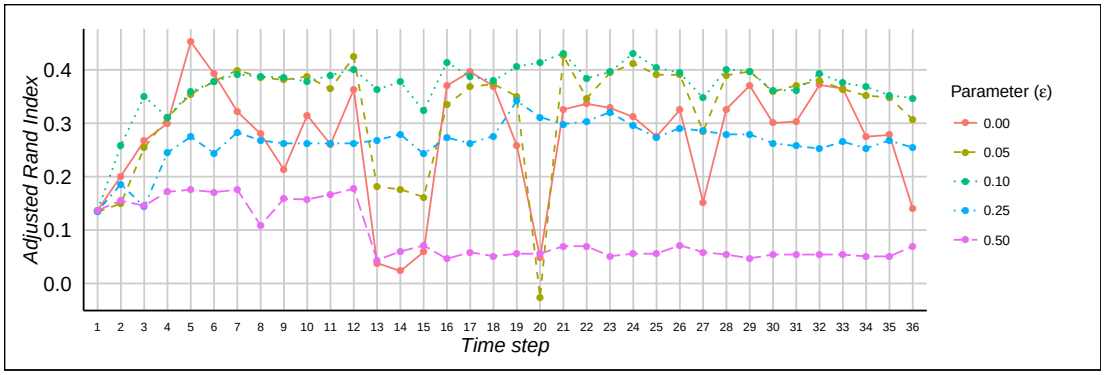


Fig. 3.6 MIT Social Evolution network - The change in Adjusted Rand Index (ARI) at each time step for SeqMod depending on the value of ε.

3.3.4 Brazilian Congress voting dataset

The Brazilian Congress network consists of 589 nodes and 12 time steps. The ground truth used as validation for this dataset is the political alignment of the congressmen, based on their parties (either left-wing or right-wing). Figure 3.7 shows the ARI results found by SeqMod at each time step for different values of ε. The best average results were found with $\varepsilon = 0.10$ and results ontained for $\varepsilon = 0.20$ are also favourable. When $\varepsilon = 0.50$, the accuracy of the algorithm decreases, but is still better than when no historical information is considered ($\varepsilon = 0.00$).

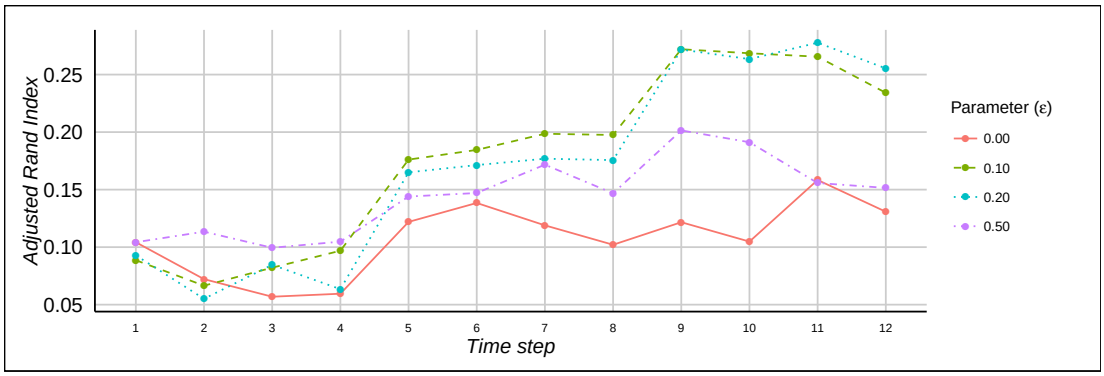


Fig. 3.7 Brazilian congress network - The change in Adjusted Rand Index (ARI) at each time step for SeqMod depending on the value of ε.

3.4 Comparative analysis

We compare SeqMod with the estrangement confinement, FacetNet, DynMOGA and genLouvain algorithms. Since estrangement and FacetNet do not output information about absent nodes and ARI only compares two membership vectors of the same size, we have assigned each of these nodes to a unique community label for comparison purposes. The parameters were set as follows: we tested ε in the range $[0.00, 0.50]$, in intervals of 0.05 for SeqMod and the maximum number of communities was set to 10, 10 and 4 for the High School, MIT Social Evolution and Brazilian Congress network,respectively. For the estrangement algorithm, the default set of parameters defined by the authors ($\delta = 0.00, 0.01, 0.025, 0.05$) were employed, plus additional parameters ($\delta = 0.1, 0.2, ..., 0.5$). For FacetNet a fixed number of communities should be specified, so we fixed it to to the number of communities in the ground truth for each dataset. GenLouvain was tested for *omega* ranging from 0.00 to 0.50 in intervals of 0.05. Across all methods, deciding on which parameter to use for comparison is important, so we report the test cases that have yielded the largest average ARI (across all time steps) for each algorithm. Table 3.2 reports the best average ARI found by each algorithm and the parameters used to find these results.

Figure 3.8 shows the results on the High School network for the parameters reported in Table 3.2 at each time step. The algorithms seem to detect a similar pattern on the network from t_3 to t_5: they all show a decrease in ARI in this period and most algorithms show an increase in this metric during the last two time steps. SeqMod, however, provided the more accurate results, matching more closely the ground truth communities of this network through time (average ARI = 0.8160).

Network	**Algorithm**								
	SeqMod		Estrangement		genLouvain		FacetNet		DynMOGA
	ARI	Parameter	ARI	Parameter	ARI	Parameter	ARI	Parameter	ARI
High School	**0.8160**	$\varepsilon = 0.15$	0.5739	$\delta = 0.00$	0.6872	$\omega = 0.20$	0.5907	$M = 5$	0.3962
Social Evolution	**0.3705**	$\varepsilon = 0.10$	0.3192	$\delta = 0.10$	0.3701	$\omega = 0.30$	0.2429	$M = 8$	0.1611
Brazilian Congress	**0.3390**	$\varepsilon = 0.10$	0.1524	$\delta = 0.01$	0.3188	$\omega = 0.10$	0.2434	$M = 2$	0.0287

Table 3.2 Best results found by each algorithm and the respective parameters that has returned such results.

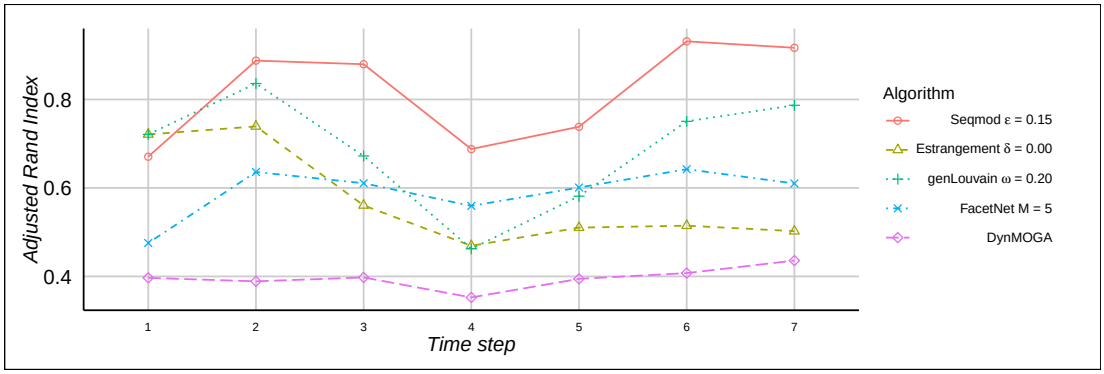


Fig. 3.8 High School network - The change in Adjusted Rand Index (ARI) at each time step for each algorithm.

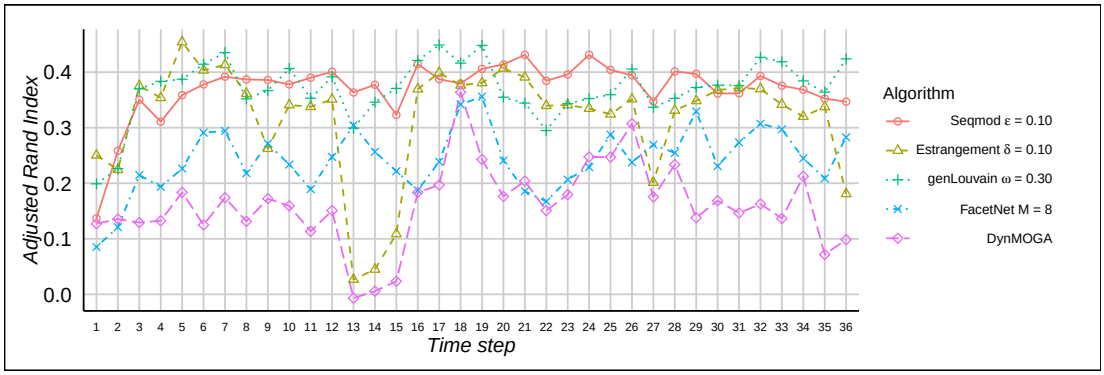


Fig. 3.9 MIT Social Evolution network - The change in Adjusted Rand Index (ARI) at each time step for each algorithm.

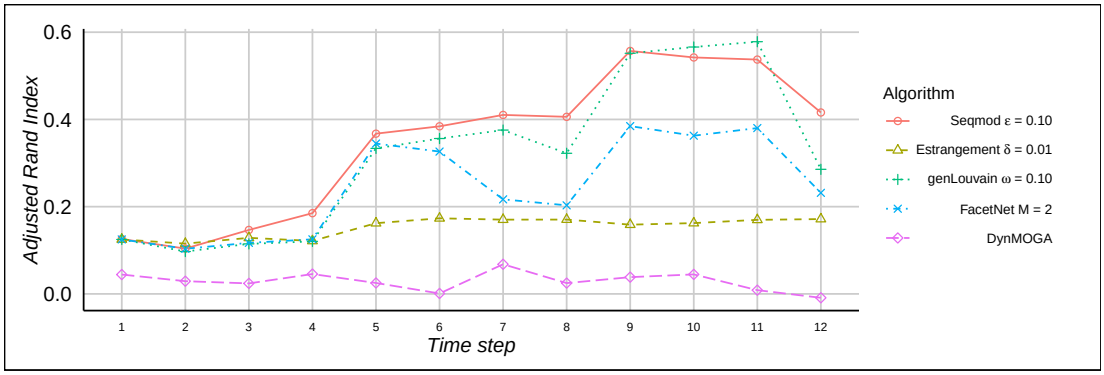


Fig. 3.10 Brazilian Congress network - The change in Adjusted Rand Index (ARI) at each time step for each algorithm.

The best results for the MIT Social Evolution network are shown in Figure 3.9. SeqMod again shows the best performance overall in terms of ARI with $\varepsilon = 0.10$ followed closely by genLouvain. As discussed previously and shown in Figure 3.6, the results found with this parameter were the closest to the ground truth. It is noteworthy that tests with smaller ε were able to show structural changes in the network during holiday periods $t_{13} - t_{15}$ and around t_{20} and t_{27}. These changes were detected by estrangement and also by DynMOGA and FacetNet, although with less accuracy.

This fact once again highlights the existing balance between detecting the ground truth of a network and the changes in network structure, which is evident even in algorithms without a smoothness parameter like FacetNet and DynMOGA. If the objective is to detect changes in the network, the smoothness parameter must be set in a way that allows for more adaptation. In SeqMod, this means that ε must be set to a lower value as discussed before. Estrangement algorithm and SeqMod with $\varepsilon = 0.05$ exhibit the more drastic discontinuities in the network.

The tests with Brazilian Congress network show similar results (Figure 3.10). SeqMod achieves the best accuracy of algorithms, followed by genLouvain. Both algorithms exhibit a sudden increase in ARI from t_4 to t_5 and from t_8 to t_9, corresponding to the years of elections and renewal of congressmen in the Chamber of Deputies in Brazil. A decrease in ARI is observed for most algorithms from t_{11} to t_{12} and since the ground truth is the alignment of the parties, we conjecture that this change reflects the changes in political alignment of the parties due to the recent political crisis in Brazil that started in 2014 when a major corruption scandal was revealed [186].

3.5 Conclusions

In this paper, we proposed an alternative approach to sequential clustering of evolving networks using mathematical programming. The proposed method is an extension of our previous work to consider historic information when detecting community structure of dynamic networks, under evolutionary clustering framework. We validated our model on dynamic networks with known ground truth communities and compared it to other methods who also employ both historic information and modularity optimisation.

Our model uses a different concept of similarity where a preservation coefficient allows for an intuitive understanding of the number of vertex pairs that were maintained in the network. We also introduce the concept of a "reference snapshot", instead of using the immediate previous time step in the historical cost, we use the previous time step with the largest modularity. The reason is that we want to maximize the ability of our model to detect the ground truth of communities through time, comparing each snapshot with a previous better defined community structure.

We have shown that our model is able to detect the ground truth of dynamic networks and changes in the network community structure. We have shown with our illustrative example that our model is able to incorporate historic information to detect probable changes in the network and that it also maintains the information, shedding light on the membership of the nodes that do not interact during all time steps, i.e. that are absent or isolated. Determining the desired smoothness is the main challenge of evolutionary clustering and consequently of our model, but from the results obtained in this study, we suggest a value around 0.10 and 0.15. Some methods automatically determine the parameter at each time step based on probabilistic models and statistical features of the

network while algorithms like DynMOGA model this process as a multiobjective problem and hence do not require any parameters, but as our tests have shown, these algorithms found poor results. We intend to improve our mathematical programming model in the future to adapt to the changes in the network without the need of parameter setting, while also providing solutions of suitable quality.

Mathematical programming provides a flexible environment for modelling community detection and here we have illustrated its use for evolutionary clustering. We note that this type of modelling framework is particularly adaptable and can accommodate various user requirements, one can for example specify a module allocation for determined nodes if known, one can add constraints that change the way isolated nodes are treated or any other logical conditions necessary for a practical application. Therefore, we believe that such approaches will prove to be popular in future network analysis methodologies and applications.

Commentary: recent updates and implications to the published work

One assumption of the work in this paper and in related work is that node attributes correspond to the real groups one could find in graphs. These metadata properties are then used to validate the network partitions discovered with metrics such as Adjusted Rand Index, used in this chapter. From this point of view, a good community detection algorithm is one that is capable to uncover these attributes from the topology of networks alone.

Some recent publications, however, have started to question the concept of "ground truth" in network partitions. Newman and Clauset [187] have presented an algorithm that takes into account the attributes of nodes to identify communities in networks but only if these attributes are correlated with a modular structure of the network. With their method, the authors have shown that certain node attributes of networks were in fact not meaningful for community detection and similar reflections can be found elsewhere in the literature [39, 188, 189].

The current validation practices for community detection algorithms are not disqualified by these observations but are likely to change in the near future. The high school network, used in this Chapter to illustrate the capabilities of SeqMod, is one example where node attributes are correlated with topological communities and the validation assumption holds true. Previous publications have confirmed that the students represented in the graph indeed tend to interact more with others who belong to the same class and with those studying similar disciplines [178].

In future works, synthetic dynamic networks with sets of node attributes correlated and uncorrelated to topological modules could be designed to provide

a better benchmark for community detection algorithms [190]. The incorporation of node attributes directly into the sequential clustering of these networks could also be used to explore the implications of the additional data into the robustness of clusters.

Chapter 4

Regression algorithms for QSAR models

Foreword

Chapter 3 of this thesis introduced an optimisation model capable of detecting the evolution of groups in networks. In this chapter, mathematical programming is explored for a different task: the prediction of biological activity of chemical compounds based on properties of their molecular structure.

These mathematical models, called Quantitative Structure-Activity Relationship (QSAR) models, have been applied over the past 60 years in studies of small sets of congeners molecules. But the recent availability of chemical databases combined with modern machine learning (ML) techniques have facilitated the analysis of larger and more heterogeneous set of molecules. Many of the predictive machine learning models, however, are "black boxes". These techniques cannot

identify a clear relationship between the molecular properties and the biological activity investigated.

The proposed algorithm attempts to overcome this issue and improve the interpretability of QSAR models. The proposed method automatically splits the chemical compounds in groups and identifies a clear relationship between features of the data and the biological outcome, producing a predictive yet interpretable model. Inheriting the benefits of mathematical programming, the model can also be easily customised according to the needs of a specific QSAR project as it is demonstrated in the results section.

4.1 Introduction

Quantitative Structure-Activity Relationship (QSAR) models are mathematical models that aim to predict biological activity of chemical compounds based on molecular structure [108]. These models are particularly useful for drug discovery as they can be used to draw hypothesis from the data, to perform virtual screening for molecules that have not yet been tested against a target of interest [109], to indicate optimisation strategies for developing new drugs from a series of promising compounds [115] or to re-purpose existing medicines to different treatments [116].

The first QSAR models were built for small series of similar compounds using linear regression with few quantitative features[122] and aimed to discover a transparent relationship between molecular structure and biological activity. Although this approach is still employed successfully to design new drugs [123, 124], most recent models consist of hundreds or thousands of molecular descriptors calculated from the chemical, 2D or 3D representations of the molecules [125–128] and are often built with non-linear algorithms such as neural networks, support

vector machines with Gaussian kernels and random forest [129] . These techniques produce highly accurate predictions but even with random forest, where a ranking of the most important features can be obtained, it is not possible to clearly identify the properties of a molecule that lead to better potency. The emphasis of these models is usually in prediction accuracy rather than in understanding how chemical structure drives biological activity [133].

It is possible, however, to produce interpretable yet accurate models for large and heterogeneous QSAR data sets by controlling either the number or type of descriptors used for modelling or by selecting a more transparent algorithm to fit the data [133, 191]. While the type of descriptors depends on specific hypothesis about the set of compounds being studied, there are a few algorithmic solutions to reduce the number of features. Principal Component Analysis (PCA) is possibly the most common of these techniques [144, 192, 193] but other supervised learning heuristics such as genetic algorithms [124], particle swarm optimisation [194] and regularisation strategies [195] have also been successfully used for variable selection. In fact, QSAR models developed with these techniques are more interpretable because they explicitly identify a subset of most relevant molecular descriptors instead of transforming the original data set.

A transparent algorithm for QSAR would also have to account for some of the non-linearities inherent to the data and yet be able to relate the contribution of the most relevant molecular descriptors to the prediction of biological activity. In this chapter, a novel computational strategy for activity prediction is proposed with this goal in mind. Our proposed algorithm, Optimal Piecewise Linear Regression Algorithm with Regularisation (OPLRAreg), identifies different regions in the data and linear equations to describe each of these segments while incorporating an explicit feature selection with regularisation. OPLRAreg represents QSAR models using mathematical programming, a standard representation of optimisation

problems that can be solved using exact algorithms and can be easily adjusted by the addition of custom constraints [196–198].

The algorithm is tested on data sets of compounds compiled from ChEMBL [199] to predict the inhibitory concentration ($logIC_{50}$) following best practices in QSAR modelling for data cleaning, preprocessing and rigorous validation [108, 200]. In the results section, it is shown how the proposed algorithm could be easily modified to accommodate custom constraints of a QSAR project and the effect of regularisation in prediction accuracy and dimensionality reduction is demonstrated. The algorithm is also compared to other machine learning algorithms found in R package caret [201] version 6.0-76 (2017).

4.1.1 Data Sets

Five data sets were obtained from ChEMBL database, using the same endpoints used to benchmark algorithms in [129]. Each data set contains a list of chemical compounds along with their respective inhibitory activity against a common target measured by the IC_{50} value, the concentration of a compound that results in 50% inhibition of the maximal activity [202]. To perform the regression analysis, the log value of IC_{50} is more frequently used and is defined as $pIC50 = -log_{10}(IC_{50})$, indicated by the PCHEMBL_VALUE column on ChEMBL.

An updated list of the chemical compounds from each of the datasets was downloaded from the ChEMBL website and a pre-processing step was performed in order to remove invalid or duplicate compounds. First, only the compounds that had their IC_{50} measurements taken were selected and then compounds with dubious measurements (indicated by column DATA_VALIDITY_COMMENT in ChEMBL) were removed. Some compounds appeared more than once in the

data set with different measurement values and were treated in a special way. Duplicate entries where the standard deviation of IC_{50} values was greater than 1, $sd_{IC_{50}} > 1$ were removed from the data set. In cases where $sd_{IC_{50}} \leq 1$, only one entry was kept to represent this chemical in the data set and its IC_{50} value was given by the median of all its duplicate entries. The Java Chemistry Development Kit (CDK) version 1.5.13 [138] and its R interface [203] was used to calculate 1D and 2D molecular descriptors for the compounds, providing 200+ numerical values to describe each structure. Mean centering was applied to these descriptors and they were placed on a scale of 0 to 1. Finally, highly correlated descriptors and those with a near zero variance in their distributions were removed from the data set using the R package *caret* [201].

Table 4.1 Summary of data sets used in this study, comprising all samples tested as inhibitors against a common drug target

Target	**Biological Endpoint**	**Source**	**Samples**	$pIC50$ **range**
NPYR1	human neuropeptide Y receptor type 1	CHEMBL4777	354	[4.12, 10.70]
NPYR2	human neuropeptide Y receptor type 2	CHEMBL4018	374	[4.01, 10.15]
CHRM3	human muscarinic acetylcholine receptor M3	CHEMBL245	588	[4.00, 10.50]
hDHFR	human dihydrofolate reductase	CHEMBL202	542	[4.00, 9.41]
rDHFR	rat dihydrofolate reductase	CHEMBL2363	875	[4.00, 9.49]

A summary of the data sets after this preprocessing step can be seen on Table 4.1. Since all compounds that had their experimental IC50 values registered on ChEMBL were retrieved to compose these data sets, the number of samples in this study range from 354 (NPYR1) to 875 (rDHFR), a large number for QSAR models. Also, all data sets have a wide range of biological activity extending from inactive compounds in the micromolar range of concentration ($pIC50 \approx 4$) to

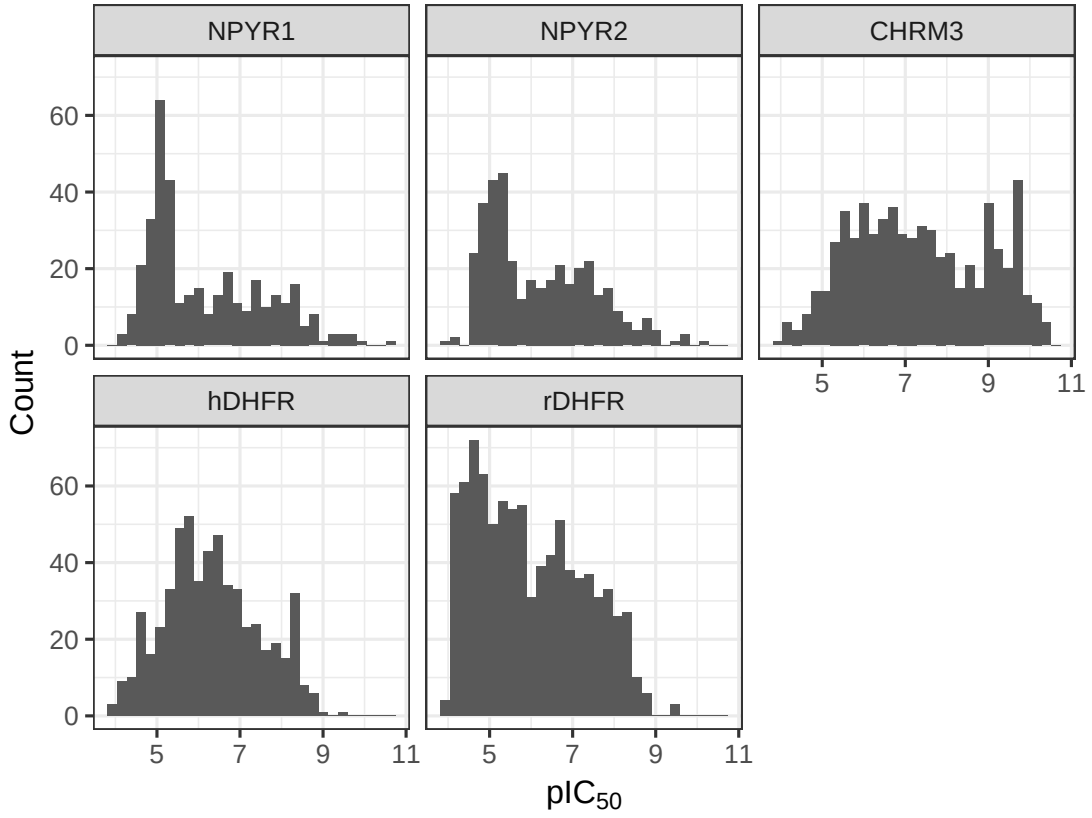


Fig. 4.1 Distribution of activity, measured by the logarithm of the 50% inhibitory concentration (IC_{50}) of compounds in the studied data sets.

potent compounds with a nanomolar activity concentration ($pIC50 \approx 10$), shown in Figure 4.1.

A background about the biological function of the target proteins and the inhibitors in the data sets are presented below.

Neuropetide Y receptors

Neuropeptide Y (NPY) is a 36 amino-acid peptide abundant in the brain and involved in various physiologic roles in the body and is a potential drug target for many disorders. The concentrations of the neuropeptide in the brain were found to be elevated in patients with eating disorders and suggests that NPY

is involved in food intake and energy expenditure. NPY is also involved in the control of the circadian rhythm, of seizures and heart neural activity. Also, its sedative properties suggests it plays a role in alcoholism and the pathophysiology of stress, major depression disorder and anxiety [204, 205]. The biological effect of NPY is mediated through the activation of its receptor subtypes. In this study, compounds tested as possible inhibitors of two NPY receptors were collected: Y1 and Y2, here indicated as NPYR1 and NPYR2 data sets, respectively. Y2 have been identified in some studies as a more promising drug target, this G-protein coupled protein is expressed in many parts of the brain and more abundant [206] and many selectivity studies have been carried out to identify small molecules with high affinity for Y2 instead of other receptors such as Y1.

Muscarinic acetylcholine receptor M3

Muscarinic acetylcholine M3 is one of five G protein-coupled receptors (M1-M5) that binds the neurotransmitter acetylcholine. The M3 receptor is encoded by gene CHRM3 and is involved in muscle contraction, glandular secretion and in the regularisation of food intake [207]. Over-expression of M3 has been linked to colon [208] and gastric cancer [209], making it an important drug target.

Dihydrofolate reductase

The enzyme dihydrofolate reductase (DHFR) plays a central role in DNA biosynthesis and cell replication [210]. Because the inhibition of DHFR leads to cell death, the enzyme has become one important drug target for the treatment against several infections such as pneumonia, toxoplasmosis, *Candidas albicanis* [211, 212], *Mycobacterium avium* complex [213], *Plasmodium falciparum* as well as cancer

[214, 215]. In this study, compounds tested for inhibition against rat and human models were collected composing data sets rDHFR and hDHFR, respectively.

4.1.2 New mixed integer programming model

A piecewise linear regression algorithm based on mathematical programming was introduced in [216]. Optimal Piecewise Linear Regression Algorithm (OPLRA) solves Mixed Integer Programming (MIP) models to find partitions in the data set so that the outcome of samples are predicted by unique linear equations identified for each disjoint region. The algorithm contains a loop defined over all features in the data set where MIPs are solved for two regions ($R = 2$), and the feature corresponding to the smallest error in prediction across all samples is taken as the partition feature (f^*) for subsequent iterations. The number of regions is then increased at each iteration until the improvement in prediction error is negligible between iterations.

Although OPLRA has been successfully applied to UCI benchmark data sets, it did not perform well when applied to QSAR models. The regression coefficients identified by the algorithm fit samples in the training set well but had a poor performance on the test set, in what was a sign of overfitting. To mitigate these problems, OPLRA was modified to account for both the accuracy and complexity of the models generated. Instead of the sum of absolute errors, prediction accuracy is now measured by mean absolute error (MAE) and a ℓ_1 regularisation term (REG) consisting in the sum of absolute regression coefficients was added to the objective function to reduce the risk of generating linear equations that are too specific to the training set. The new objective function in Equation 4.1 below.

$$z = MAE + \lambda\, REG, \tag{4.1}$$

where λ is a user defined parameter that controls the influence of regularisation.

Variables MAE and REG are defined by the set of equations below:

$$MAE = \frac{\sum_s E_s}{|s|}, \tag{4.2}$$

$$REG = \sum_f W_f^+, \tag{4.3}$$

$$W_f^+ \geq \quad W_f \qquad\qquad \forall f \tag{4.4}$$

$$W_f^+ \geq -W_f \qquad\qquad \forall f, \tag{4.5}$$

where E_s indicates the absolute error for each sample s and $|s|$ is the number of samples in the training set. Positive variables W_f^+ were introduced to indicate the absolute value of regression coefficients W_{rf} and are defined by the two auxiliary constraints above.

At every iteration, the number of regions R and the partition feature f^* used to identify breakpoints are fixed. The allocation of sample s to regions $r \in \{1, 2, \ldots, R\}$ is modelled with binary variables F_{sr} while the breakpoints are represented by the free variables $X_{r,f}$, where f always correspond to the partition feature f^* of the current iteration.

Equation 4.6 guarantees that a sample can belong to only one region:

$$\sum_r F_{sr} = 1 \quad \forall s, \tag{4.6}$$

while Equation 4.7 below makes sure that the breakpoints are consistent.

$$X_{r,f^*} \geq X_{r-1,f^*}, \quad \forall r = 2, 3, \ldots, R-1, \tag{4.7}$$

Equations 4.8 and 4.9 assign samples to the correct regions according to the breakpoints.

$$A_{sf^*} \geq X_{r-1,f^*} - U\ (1 - F_{sr}) \qquad \forall s, r = 2, 3, \ldots, R, \tag{4.8}$$

$$A_{sf^*} \leq X_{r,f^*} - U\ (1 - F_{sr}) \qquad \forall s, r = 1, 2, \ldots, R - 1, \tag{4.9}$$

The predicted value P_{sr} for sample s in region r is given by Equation 4.10, according to regression coefficients W_{rf} and the intercept B_r for each region. Equations 4.11 and 4.12 compute the absolute error in prediction E_s for each sample. O_s are the observed values for sample s and U is a large number that will force these constraints to consider only the predicted values P_{sr} where sample s belongs to region r, $F_{sr} = 1$.

$$P_{sr} = \left(\sum_f W_{rf} A_{sf}\right) + B_r \qquad \forall s, r, \tag{4.10}$$

$$E_s \geq O_s - P_{sr} - U\ (1 - F_{sr}), \qquad \forall s, r, \tag{4.11}$$

$$E_s \geq P_{sr} - O_s - U(1 - F_{sr}), \qquad \forall s, r, \tag{4.12}$$

The full MIP model is then given by:

$$\begin{aligned} &\text{minimise} \quad z \\ &\text{subject to} \\ &\qquad \text{Equations} \qquad (4.1) - (4.12) \end{aligned}$$

Implicit feature selection

Regularisation reduces the risk of overfitting in the model but it can also be used as tool for feature selection. Regression coefficients of less important features will be set to zero automatically for $\lambda > 0$ because the regularisation term will have a larger influence in the optimisation process. We can take advantage of this side effect and restrict the feature space of the problem to only those with nonzero coefficients after the first iteration of the algorithm. This implicit feature selection reduces the size of MIP models in the remaining iterations as well as the number of loops required in the iteration with 2 regions.

4.1.3 Proposed algorithm

Algorithm 1 summarises the iterative process of the proposed OPLRAreg with the modifications described above. First, a simple linear regression is fit to the training data (number of regions $R = 1$) and z is recorded. The regularisation will naturally ensure that the coefficient of less relevant features are set to zero and only descriptors that have been effectively used in the linear equation are kept for the next iterations. Note that constraints related to breakpoints and assignment of samples to regions (Equations 4.7, 4.8, 4.9) are not used while solving the first MIP model and all samples are assigned to a single region, $F_{sr_1} = 1$ according to Equation 4.6. Then, a MIP with two regions ($R = 2$) is solved for each selected feature and the feature corresponding to the best model in this iteration is the partition feature f^* for the remaining iterations. The number of regions increases until the improvement of absolute error in consecutive iterations is no more than a user-defined parameter β.

Algorithm 1 OPLRAreg: Optimal piecewise linear regression algorithm (OPLRA) with regularisation

Solve OPLRAreg for R = 1 ▷ Simple linear regression
$\text{ERROR}_{\text{current}} \leftarrow z$
$\text{ERROR}_{\text{old}} \leftarrow \infty$
$\text{ERROR}_{\text{tmp}} \leftarrow \infty$
$f_{\text{best}} \leftarrow \{\}$
$F \leftarrow \{f \in \mathbf{f} \mid W_{r_1,f} \neq 0\}$ ▷ Implicit feature selection
$R \leftarrow 2$
for $i \leftarrow 1; i \leftarrow i+1; i \leq F$ **do** ▷ Selects best partition feature in 2 regions
    Solve OPLRAreg with 2 regions and partition feature f_i
    if $z < \text{ERROR}_{\text{tmp}}$ **then**
        $\text{ERROR}_{\text{tmp}} \leftarrow z$
        $f_{\text{best}} \leftarrow f_i$
    end if
end for
$\text{ERROR}_{\text{old}} \leftarrow \text{ERROR}_{\text{current}}$
$\text{ERROR}_{\text{current}} \leftarrow \text{ERROR}_{\text{tmp}}$
$f^* \leftarrow f_{\text{best}}$
while $\text{ERROR}_{\text{current}} < (1-\beta)\text{ERROR}_{\text{old}}$ **do** ▷ Number of regions increases
    $R \leftarrow R+1$
    Solve OPLRAreg with R regions and partition feature f^*
    $\text{ERROR}_{\text{old}} \leftarrow \text{ERROR}_{\text{current}}$
    $\text{ERROR}_{\text{current}} \leftarrow z$
end while
return partition feature f^*, breakpoints X_{rf}, regression coefficients for each region W_{rf}

4.1.4 Implementation and Validation scheme

The validation scheme used in this study is illustrated on Figure 4.2 and is aligned with state of the art QSAR model validation procedures [108, 200]. Data sets are initially split randomly, 75% of each data set is used to construct QSAR models while 25% is used as external validation set. The 75% set is again divided into 10 training and test folds using a stratified sampling technique available in *caret* and this is done 10 times. Therefore, with this cross-validation strategy, each algorithm produces 100 different QSAR models constructed from different subsets of data. Of these, the model that had the smallest MAE is selected and used to predict biological activity of samples in the external validation set (25%). This procedure of model selection was reproduced 5 times for each data set.

4.1.5 Comparative analysis

The following nonlinear algorithms present in *caret* were compared to OPLRAreg results: Support Vector Machine [217, 218], Random Forest [219], Neural Networks [220] and Random GLM [221]. We have also compared OPLRAreg against the following linear algorithms present in *caret*: Generalised Linear Model (GLM), Lasso, Linear Regression, Partial Least Square (PLS) and Elastic Net.

4.2 Results and Discussion

In this section, the results of the piecewise linear regression algorithm are shown. The division of regions can help elucidate details of QSAR data sets and results are compared to other machine learning algorithms.

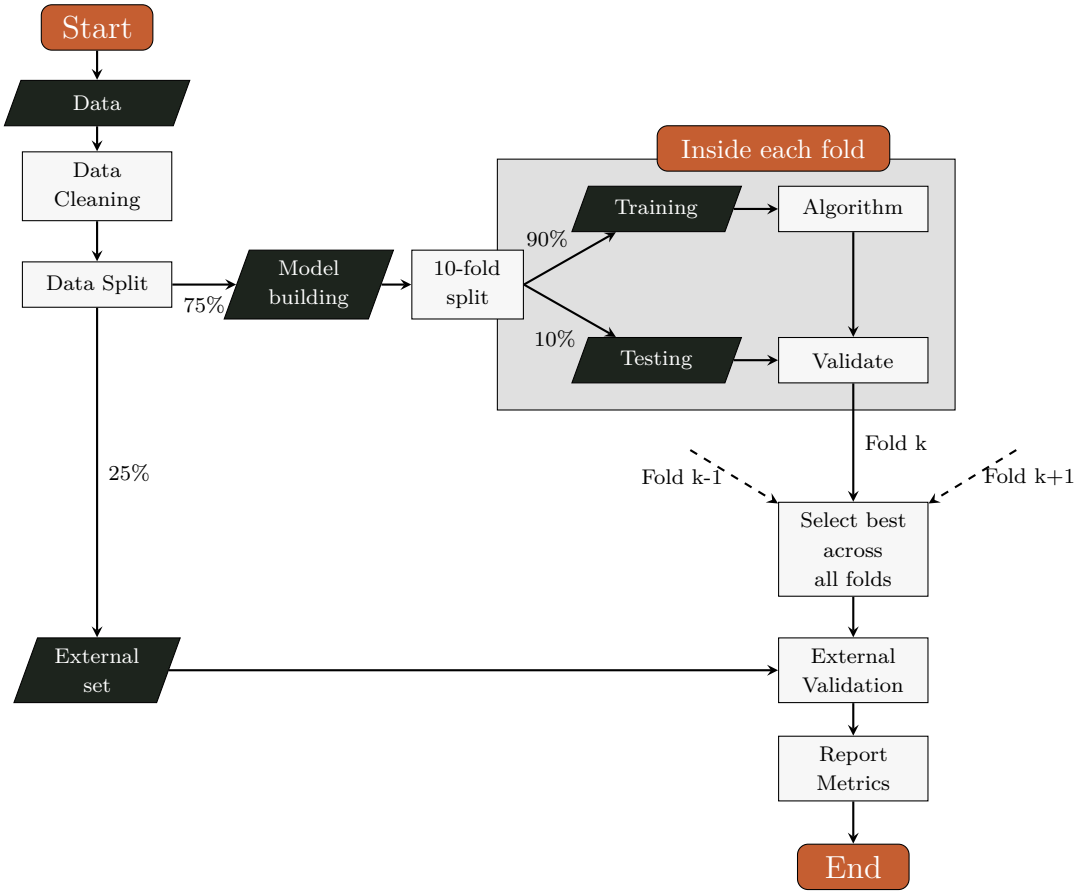


Fig. 4.2 Validation scheme adopted in this study.

4.2.1 Parameter optimisation

In initial tests to understand the impact of the regularisation parameter, the stopping criteria parameter was kept constant at the best value found in the previous study, $\beta = 0.03$ [216] while the regularisation parameter assumed one of the following values: $[0.000, 0.005, 0.010, 0.200]$. The tests, performed in a single round of 10-fold cross-validation, clearly show that performance of OPLRA is improved by the introduction of regularisation term. Table 4.2 shows mean and standard deviations of mean absolute error (MAE), CPU time required to run each test case and average number of features and regions detected by the algorithm according to the regularisation parameter used.

Table 4.2 Performance of piecewise linear algorithm for different regularisation parameters.

	rDHFR	hDHFR	CHRM3	NPYR1	NPYR2
MAE					
$\lambda = 0.000$	54.76 ± 70.31	28.56 ± 29.51	103.15 ± 118.52	124.28 ± 105.66	153.32 ± 147.09
$\lambda = 0.005$	0.74 ± 0.07	0.79 ± 0.06	0.78 ± 0.06	0.70 ± 0.09	0.57 ± 0.12
$\lambda = 0.010$	0.84 ± 0.06	0.84 ± 0.08	0.78 ± 0.07	0.76 ± 0.09	0.63 ± 0.09
$\lambda = 0.020$	0.90 ± 0.07	0.85 ± 0.05	0.83 ± 0.10	0.73 ± 0.11	0.61 ± 0.08
Time (min)					
$\lambda = 0.000$	90.84 ± 3.98	44.93 ± 3.23	60.44 ± 1.57	24.82 ± 3.84	24.59 ± 4.30
$\lambda = 0.005$	9.73 ± 0.83	4.64 ± 0.73	7.40 ± 0.60	4.70 ± 0.78	5.38 ± 0.62
$\lambda = 0.010$	5.41 ± 0.93	1.86 ± 0.36	5.15 ± 0.71	3.69 ± 1.59	2.82 ± 0.39
$\lambda = 0.020$	2.17 ± 0.78	0.62 ± 0.14	2.48 ± 0.21	2.31 ± 0.23	1.55 ± 1.59
Features					
$\lambda = 0.000$	80.0 ± 0.00	75.9 ± 0.32	86.2 ± 0.42	69.2 ± 0.42	67.0 ± 0.00
$\lambda = 0.005$	21.9 ± 1.60	19.9 ± 1.80	23.7 ± 1.57	22.4 ± 2.80	25.1 ± 3.41
$\lambda = 0.010$	13.4 ± 1.43	8.9 ± 2.69	16.8 ± 1.81	16.4 ± 2.12	14.7 ± 1.83
$\lambda = 0.020$	5.0 ± 0.67	2.6 ± 0.52	12.0 ± 2.26	9.4 ± 0.97	7.3 ± 0.48
Regions					
$\lambda = 0.000$	4.3 ± 0.82	4.4 ± 0.97	4.0 ± 0.47	4.8 ± 1.03	4.8 ± 1.87
$\lambda = 0.005$	2.0 ± 0.00	2.0 ± 0.00	2.0 ± 0.00	2.3 ± 0.48	2.0 ± 0.00
$\lambda = 0.010$	2.1 ± 0.32	2.0 ± 0.00	2.0 ± 0.00	2.3 ± 0.95	2.0 ± 0.00
$\lambda = 0.020$	2.1 ± 0.32	2.0 ± 0.00	2.0 ± 0.00	2.0 ± 0.00	2.3 ± 0.95

As mentioned in the previous section, OPLRA had a poor predictive performance on QSAR data sets. The predicted variable in most datasets ranges from

$pIC_{50} = 4$ to $pIC_{50} = 11$ but the mean absolute error in the tests without regularisation ($\lambda = 0$) went well above this range. The best regularisation parameter value was found to be $\lambda = 0.005$, where prediction accuracy was consistently better on all data sets when compared to tests with nonzero λ. OPLRAreg was also 4 to 10 times faster with the optimal regularisation parameter and the average number of features selected was around 20.

4.2.2 Algorithm results

On average, OPLRAreg detects 3 regions and selects from 20 to 25 features for the QSAR data sets used in this study, as shown in Table 4.3. Examples of QSAR models generated by the algorithm for data sets hDHFR and NPYR1 can be seen in Figures 4.3 and 4.4, respectively. The distribution of scaled descriptor values for the partition feature is shown against biological activity (pIC_{50}) as well as breakpoints and equations detected for each region.

Table 4.3 Average number of regions and selected features found by OPLRAreg during cross-validation

	rDHFR	hDHFR	CHRM3	NPYR1	NPYR2
Regions	3.10 (±0.31)	3.00 (±0.00)	3.00 (±0.06)	3.46 (±0.58)	3.04 (±0.19)
Features	22.30 (±2.24)	18.93 (±2.13)	25.53 (±2.50)	22.66 (±2.58)	24.95 (±2.89)

In the first example, shown in Figure 4.3, the partition feature is *MDEN-11*, a descriptor related to the distance between all primary nitrogen atoms in the molecular graph. Most samples in this data set have either MDEN-11 = 0 (23.4%) or MDEN-11 = 0.43 (71.96%) and OPLRAreg captures different equations for those cases. The algorithm assigns molecules without nitrogen atoms or with small distance between these atoms to Region 1, another multiple linear

relationship encompassing samples in $0.17 \leq$ MDEN-11 < 0.72 and it estimates that $pIC_{50} = 5.04$ for the few cases where *MDEN-11* is large. Most selected features are related to topological characteristics of the molecules and are either related to connectivity of atoms (topoShape, MDEN-22, MDEC-23, C1SP3, C3SP3, SC-5, SCH-5) or to the number of specific groups found in the molecules, as is the case of nE (number of glutamic acid) and fragments identified as Kiers Hall Smart descriptors (*khs-*) [144].

Similarly, we can interpret the breakpoints and equations for NPYR1 shown in Figure 4.4. *C1SP3* is the partition feature and it represents the number of singly bound carbon atoms bound to one other carbon. Descriptors are scaled during preprocessing of the data and the interval $[0, 1]$ represents the original range $[0, 41]$. Therefore, molecules with at most 4 such types of carbon ($C1SP3 \leq 0.11$) are predicted by equation in Region 1 while those ranging from 4 to 11 atoms belong to Region 2. Region 3 captures the rare cases (only 8% of the samples) where molecules have more than 11 carbons with the defined connectivity.

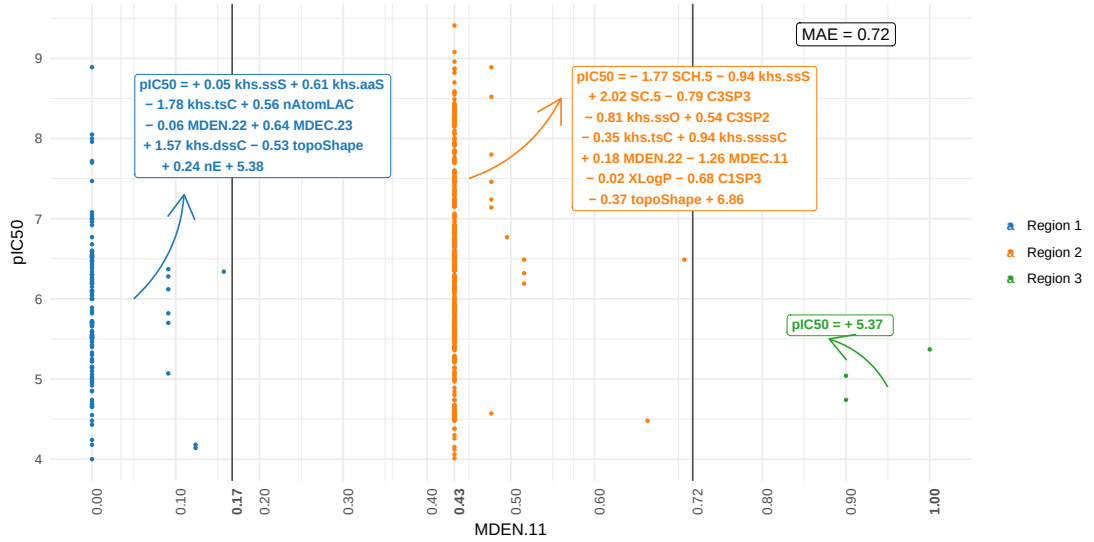


Fig. 4.3 Breakpoints, regions and equations found by OPLRAreg for data set hDHFR

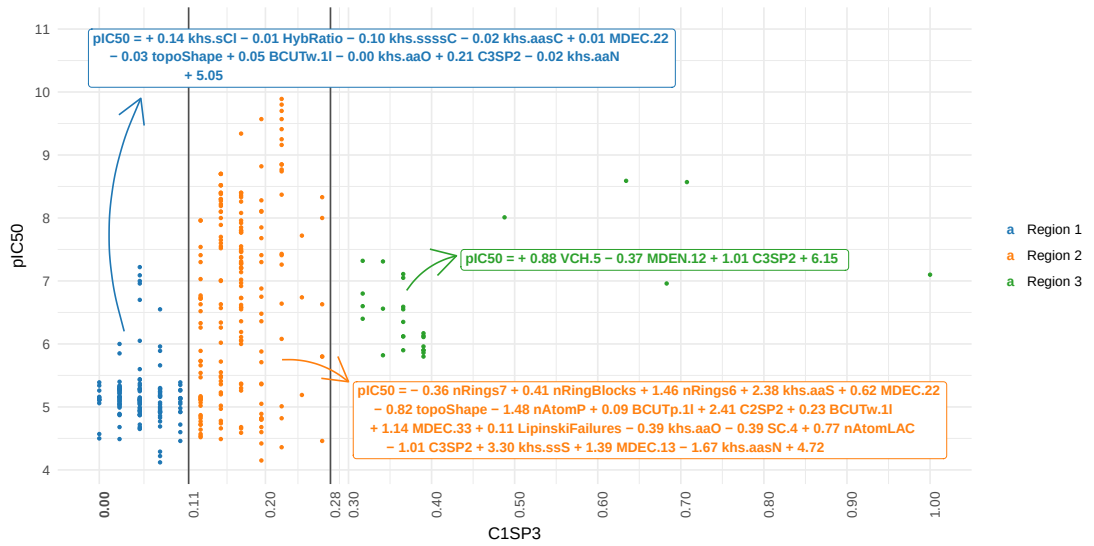


Fig. 4.4 Breakpoints, regions and equations found by OPLRAreg for data set NPYR1

4.2.3 Overall Variable Importance

One simple way to identify the most important descriptors of OPLRAreg models is to count how many regions selected a specific feature to form a multiple linear equation but that would not account for the number of samples in each region. Take the model in Figure 4.4 as an example. Descriptors *VC-5* and *MDEN-12* would be attributed a score as high as features occurring only once in Regions 1 and 2, for instance *nRings7* or *nAtomLAC*, but these features only describe the relationship of a minority of samples.

A better approach perhaps is to calculate the percentage of samples in the data set that have been predicted by equations containing a specific feature. This percentage was computed for each feature in the best OPLRAreg models selected after cross-validation and averaged across the 5 data splits to generate an overall importance score for these tests. Table 4.4 shows the top 15 features ranked according to this score per data set and the types of descriptors more frequently selected are briefly described next.

Table 4.4 Top 15 features and their importance score for each data set

Rank	rDHFR		hDHFR		CHRM3		NPYR1		NPYR2	
	Descriptor	Score	Descriptor	Score	Descriptor	Score	Descriptor	Score	Descriptor	Score
1	VC.5	98.86	khs.aaNH	99.45	MDEC.33	98.21	SC.6	99.15	SC.4	99.73
2	ALogP	95.94	VP.7	99.45	BCUTc.1h	98.19	BCUTw.1l	91.71	MDEO.11	99.47
3	MDEN.13	93.90	khs.ssS	94.60	BCUTw.1l	98.13	C3SP2	77.01	khs.ddssS	96.47
4	MDEC.22	91.77	topoShape	93.95	nG	98.13	khs.aaO	74.01	C3SP3	92.35
5	SCH.6	85.03	ALogp2	87.55	VCH.6	98.13	khs.aaS	71.98	LipinskiFailures	91.71
6	MDEC.33	84.80	khs.aaN	87.55	ATSm1	97.45	C3SP3	70.90	BCUTp.1h	91.51
7	MDEC.13	82.26	MDEN.22	84.19	khs.aaaC	97.45	MDEC.12	70.20	C3SP2	91.44
8	khs.ssNH	81.01	XLogP	78.93	nF	96.77	nAtomLAC	64.27	khs.aaO	91.44
9	ALogp2	80.11	MDEC.22	78.78	nRings6	94.39	LipinskiFailures	64.24	HybRatio	91.31
10	C1SP3	76.80	MDEN.11	78.78	MDEN.33	92.86	nRings6	62.29	khs.sF	91.31
11	nRings6	75.82	nBase	78.78	LipinskiFailures	91.84	SC.4	62.15	khs.ssO	86.36
12	tpsaEfficiency	73.66	LipinskiFailures	77.21	khs.dsCH	91.04	khs.aaaC	61.58	tpsaEfficiency	86.36
13	BCUTc.1l	72.69	MDEO.22	76.66	khs.dssC	90.22	MDEO.11	61.30	BCUTc.1l	83.69
14	khs.aaaC	72.39	C3SP2	75.89	khs.ssNH	89.56	khs.aasN	60.85	MDEN.22	83.69
15	khs.dssC	72.39	C1SP3	75.83	ALogp2	89.20	khs.sCl	60.34	ATSc3	82.80

Fragment count: Descriptors that represent the number of specific fragments or substructures. Of these, Kiers Hall Smart descriptors [143, 144], identified by the prefix *khs*, were selected more often and had a high score of importance in OPLRAreg models.

- *khs-** descriptors
- nRings6
- nBase
- nAtomLAC
- Aminoacids count (nG, nF)

MDE descriptors: Molecular Distance Edge descriptors represent the distance edge between specific atom types in the molecular graph. MDEO.11 and MDEO.22, for example, calculate the distance between all primary oxygen and all secondary oxygen, respectively.

- MDEN.11
- MDEN.13
- MDEN.22
- MDEN.33
- MDEC.12
- MDEC.13
- MDEC.22
- MDEC.33
- MDEO.11

Carbon connectivity: Descriptors describing carbon types.

- C1SP3
- C3SP2
- C3SP3

Log P descriptors: Descriptors related to the lipophilicity of molecules, an important property determinant of the absorption, transport and excretion of a drug. The logarithm of the partition coefficient, log P, can be approximated by various numerical methods:

- ALogP
- ALogP2
- XLogP
- MLogP

BCUT descriptors: Descriptors based on eigenvalues of a matrix representation of the molecular graph where diagonal weights contain either atomic **w**eight, partial **c**harge or **p**olarizability properties of molecules.

- BCUTc.1l
- BCUTc.1h
- BCUTw.1l
- BCUTp.1h

BCUT descriptors condense a great deal of information and are harder to interpret as they cannot be linked directly to properties in the molecular graph like fragments, atom types and distances descriptors. However, these descriptors have been proved useful in QSAR models as representative features of the ligand-receptor interactions [145]. A possible workaround to interpret QSAR models where these features have been deemed important is to complement the analysis of BCUT values with other correlated descriptors and visual data exploration [222].

4.2.4 Custom constraints to the model

In the previous sections, it was shown that OPLRAreg automatically finds a feature to split the data into regions. But there might be a number of equally optimal solutions, each dividing the data using different group arrangements and different partition features. Suppose that we want to discover the possible structure-activity relationships of inhibitors for a particular attribute of interest or suppose we have reasons to believe that the data can be split into specific number of regions. The proposed method is flexible enough to accommodate these requirements. The algorithm can identify an optimal division for the data

according to a predefined partition feature and output the linear combinations for each defined group.

For example, Figure 4.5 shows an alternative optimal piecewise model for hDHFR inhibitors identified by OPLRAreg when the partition feature was defined beforehand as $f^* = \text{khs.aaNH}$. This model splits the data in only 2 regions by the breakpoint khs.aaNH = 0.49, which in practice separates the compounds containing the fragment defined by the khs descriptor (khs.aaNH = 0) from those without this fragment. The accuracy of the new model (MAE = 0.74) is very similar to the one identified by the standard workflow in Figure 4.3 (MAE = 0.72) and the selection of one over the other would depend on the practical applications of this QSAR model.

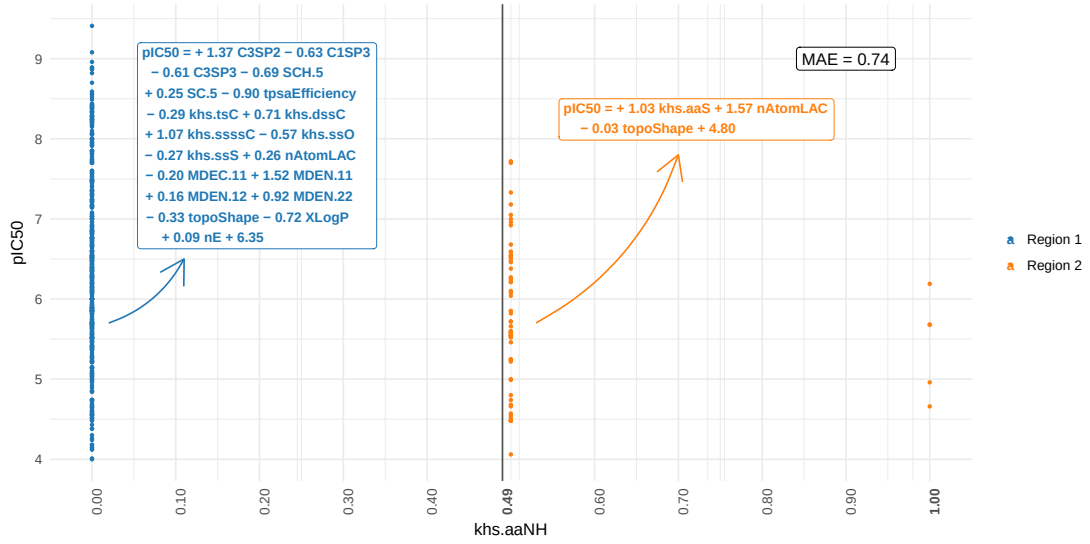


Fig. 4.5 Piecewise model for hDHFR inhibitors with khs.aaNH as the partition feature

All of these modifications to the standard procedure presented in Algorithm 1 are valid and even encouraged when developing QSAR models with OPLRAreg but they require a prior knowledge about how the data are structured. If custom constraints are specified, the algorithm will require fewer iterations and fewer MIP models to solve and as a consequence, OPLRAreg will run faster than the

original workflow. When a partition feature is informed, the initial loop that searches for the optimal region (lines 8-14 in Algorithm 1) will not be executed and if the number of predefined regions is small and also specified beforehand, only one MIP model will need to be solved. However, one must be careful with the personalisation of the model, a poor selection of the partition feature might produce inaccurate models and large number of regions will tend to overfit the model, leading to poor generalisation.

4.2.5 Comparison with other algorithms

OPLRAreg was compared to other machine learning algorithms available through R package *caret*, following the validation scheme shown in Figure 4.2. All algorithms were trained on the same train/test splits of data for 10-fold cross-validations (CV) repeated 10 times. The best models (smallest MAE on internal test set) found during CV were then used to predict samples on the external validation set. OPLRAreg parameters were set to $\lambda = 0.005$ and $\beta = 0.03$, the default parameters were used for RGLM ($nBags = 100$ and default settings for $nFeaturesInBag$) and parameters for other algorithms in *caret* package were defined by grid search, same as used in [200]. This process was repeated 5 times and the aggregated results for the external validation sets are shown in Figure 4.6 for NPYR1, NYPR2, CHRM3 and hDHFR datasets and in Figure 4.7 for dataset rDHFR.

OPLRAreg produce models that have an accuracy comparable to state of the art algorithms such as Random Forest, SVM Radial, Neural Networks and Partial Least Squares. The average error of these algorithms is around ± 0.60, the expected error for biological activity reported in ChEMBL [129]. The proposed algorithm, however, has a more interpretable and transparent output and can be easily customised, as demonstrated in the sections above. The large number

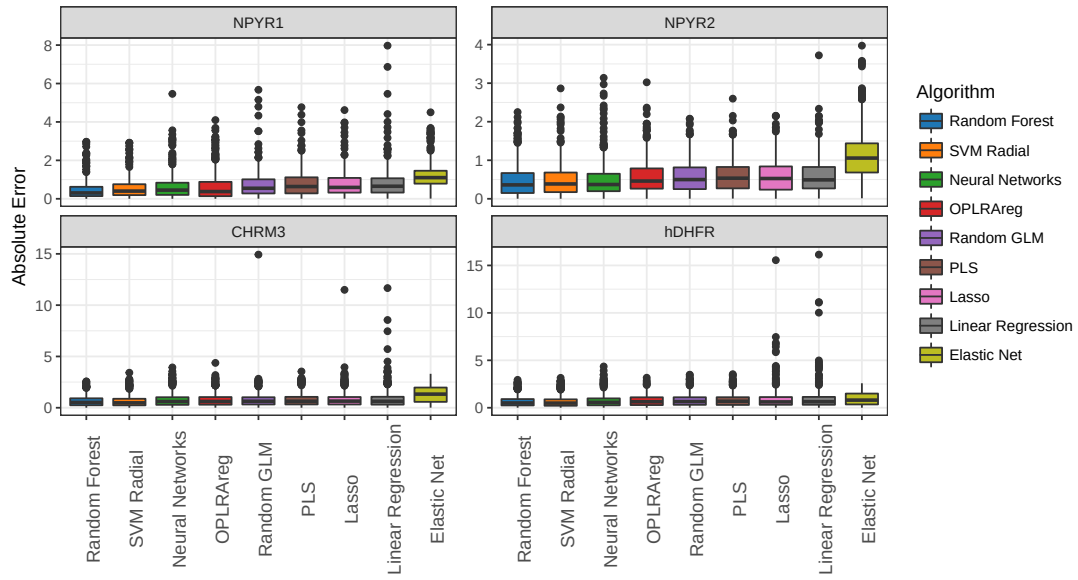


Fig. 4.6 Comparison of OPLRAreg to other machine learning algorithms

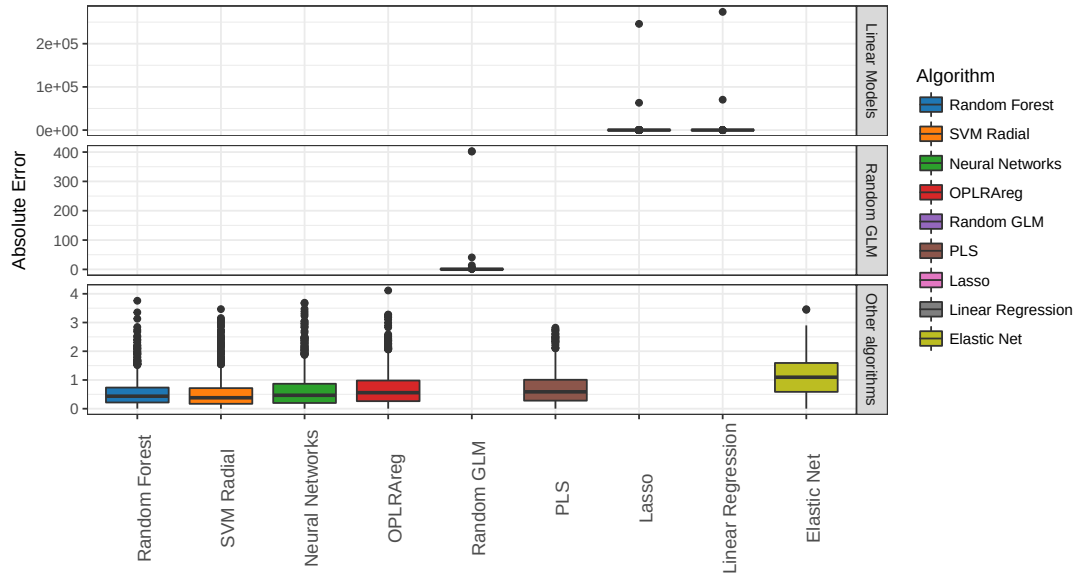


Fig. 4.7 Comparison of OPLRAreg to other machine learning algorithms (Dataset rDHFR)

of trees in a Random Forest, the vectors of SVM and the weights of neurons in Artificial Neural Networks cannot be directly interpreted and linked to features of the molecules, but models with OPLRAreg consist in simple rules and equations that are transparent and directly correlate descriptors with activity.

The box plots also show that all methods make some erroneous predictions of 2-5 orders of magnitude, these outliers are probably due to the inherent limitations of QSAR models, such as the heterogeneity of these larger data sets and the presence of activity cliffs. Models built with Linear Regression, Lasso and in some test cases, Random GLM, were good on average but produced more outlier and erroneous predictions. In test cases for data sets CHRM3 and hDHFR, it is possible to see that these models have made predictions with errors of more than 10 orders of magnitude. More noticeably, these algorithms also had a poor performance in rDHFR test case, where very large outliers can be seen in the prediction (Figure 4.7). These results suggest that these methods are more prone to overfit the training data and produce wrong predictions on data in the external set. QSAR models built with Elastic Net had a consistently larger error than average but a range of outliers consistent with the best performing algorithms.

4.3 Conclusions

In this study, the application of a piecewise linear regression algorithm based on mixed integer programming models to develop predictive QSAR models is proposed. A feature selection model was added and included a regularisation term to the objective function of a previous algorithm and the method was tested under a robust validation scheme to predict biological activity of chemical compounds against a common target. The new approach produces interpretable models and with an accuracy similar to state of the art techniques in the field.

Interpretability is one of the major drawbacks of most black box machine learning algorithms, and these improvements over OPLRA marks the first step towards making this algorithm useful for QSAR models, providing models that

are more easily interpretable. The proposed method splits the data into disjoint regions and help to explain differences in potency between different subgroups of the data by the suitable selection of features for each region. This transparent and automatic division of the data set can be useful to identify groups of compounds in early stages of drug discovery, where a large number of compounds are tested for their potency. During lead optimisation, an expert could also explore these groups and propose new molecules based on the descriptors selected for each compound series.

From the mathematical modelling perspective, OPLRAreg could be improved in the future by introducing more nonlinear transformations in the algorithm. Even though OPLRAreg already accounts for some non-linearities in the data because of the group division, it is likely that some descriptors do not correlate linearly with biological activity inside the regions defined. One simple extension would be to introduce automatic variable transformation, where descriptors could be transformed by nonlinear functions and the optimisation model would select the best transformation in each scenario. This has already been implemented elsewhere [194] but using a different optimisation technique, particle swarm optimisation, and tested on a smaller data set.

In future works, it would also be interesting to test other techniques to split the data set. Instead of a single partition feature, the algorithm could identify multiple features that separate the data better or perhaps an aggregation of features. An appropriate mathematical model would have to be created for that and it would be important to keep the model as interpretable as possible. The work described in the next chapter is one extension of OPLRAreg and is connected to this idea of alternative group representation. There, the overall similarity between molecules, represented as a network, is used to define different groups for the same data set used in the current chapter.

Chapter 5

Predictive QSAR models incorporating chemical networks

Foreword

This Chapter explores an alternative clustering strategy to separate molecules into groups using a network representation of chemical compounds. The piecewise linear model described in Chapter 4 made accurate predictions of biological activity compared to state of the art algorithms but the division of data was based on the values of a single molecular descriptor. Although this simple division strategy was capable of generating predictive rules, dissimilar molecules were often put in the same group because they shared a single topological feature or fragment; the clusters did not consider the overall similarity between molecules.

In the network based algorithm presented in this chapter, molecules are linked according to their structural similarity and molecules are separated in a hierarchical partitioning scheme. At the top layer, molecules are divided into groups identified

with modularity optimisation, which guarantees that the most similar molecules will be clustered together. Then, each of these modules are further clustered and predicted by the piecewise equations of the OPLRA algorithm introduced in the previous chapter.

The aim of the work developed and described in this chapter was twofold: enhance the capabilities of OPLRA, by creating a more informative division of the data sets and explore the role of network analysis and community detection for QSAR modelling, something still little explored in the QSAR literature. The result was a more predictive algorithm with more intuitive and interpretable subgroups. This chapter describes how these networks of compounds are created, their topological properties, it discusses the meaning of the modules and the predictive capabilities of the proposed algorithm.

5.1 Background

The topic of network analysis has been gaining attention in the drug discovery community lately [223]. Molecular networks, or chemical space networks, can help identify previously unknown drug side effects, re-purpose existing drugs to different diseases, identify mechanisms of action, as a visual aid to help elucidate structure-activity relationship and to explore activity cliffs [157, 223, 158, 224]. These investigations are usually performed as an exploratory or as a post-processing step of a QSAR model [108].

Representing molecules as a network requires a metric of similarity between the compounds and the selection of a threshold t_α. To calculate similarity, molecules are commonly encoded as fingerprints, binary strings of a fixed length. One example is the Extended Connectivity Fingerprint 4 (ECFP4), a fingerprint

technique that generates an array of 1024 bits using features of the neighbourhood of atoms in the molecule within 4 atoms of distance [225]. Pairs of fingerprints are then compared using the Tanimoto coefficient (Tc) [226], yielding a value from 0 to 1 indicating the degree of similarity between two compounds. To finally build the network, a threshold t_α is applied so that all matrix entries below this cut-off value are set to 0 and those above are set to 1, creating an unweighted undirected network.

The appropriate selection of a threshold value is important for the study of structure-activity relationship. If the threshold value is too low, the similarity matrix between molecules is too dense, making it difficult to mine important relationships. On the other hand, a threshold set too high creates a disconnected network that might not grasp the real meaning of similarity among compounds [223]. There is no consensus, however, about the best threshold and the selection of the value depends on the application and on the fingerprint used [155]. For networks built with ECFP4, $t_\alpha = 0.30$ is usually applied if at least remote structural similarity is to be represented [158, 227] and values around 0.50 or 0.60 can be found in the literature if only the most similar compounds are to be connected [228, 154]. The threshold can also be defined according to a desired edge density in the network [224].

Recently, [229] have demonstrated that the best modules of molecular networks are found with the threshold corresponding to a peak in average clustering coefficient (ACC). ACC is maximum (ACC = 1) when all nodes are connected to each other ($t_\alpha = 0$) but as the threshold is incremented, ACC tends to decline until it increases again to a peak in the interval $0.20 < t_\alpha < 0.40$. Larger thresholds show a continuing decline in clustering coefficient and although a few other local spikes can be found, none lead to larger ACC than the initial peak. The authors have shown that the most realistic modular structure can be uncovered by a

community detection algorithm when the threshold selected corresponds to a value near the initial peak.

Communities in networked data can be explored by machine learning algorithms to make predictions and classify the data [230–232, 198]. In the QSAR community, networks have been used to investigate groups in the data and activity cliffs but their community structure have not been directly explored to create predictive algorithms. A new method, Modular (OPLRA), is proposed in this chapter to try to fill this gap. The algorithm takes advantage of the properties of networks built from QSAR data sets to identify an initial grouping of data sets and the transparency and flexibility of OPLRAreg to build piecewise models, which further divide the data into regions.

In the following sections, a initial exploratory data analysis of QSAR network properties is presented, followed by the introduction of the hierarchical and modular approach mentioned above.

5.2 Network analysis

5.2.1 Network construction

The construction of networks for QSAR datasets used in this study (Table 4.1) were analysed under the optimal threshold procedure. We used CDK 1.5.13 [138] to generate ECFP4 fingerprints and calculate Tanimoto coefficient and igraph to calculate the correlation coefficient of networks generated by each threshold in $t_\alpha = 0$ to $t_\alpha = 1$ incremented by 0.01. The peak in average clustering coefficient (ACC) described in [229] was noticed in all data sets studied.

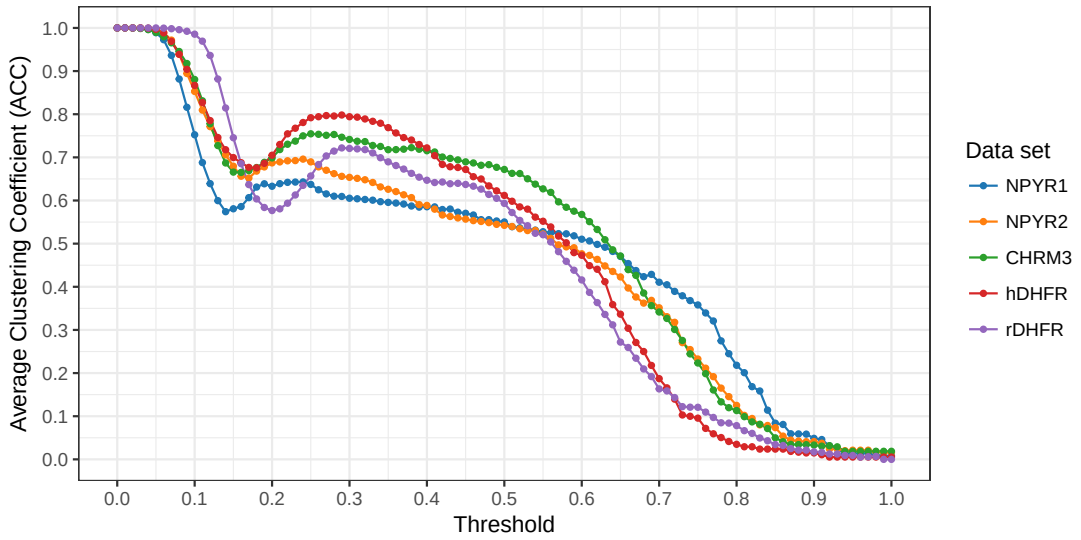


Fig. 5.1 Threshold analysis for network representation of QSAR datasets used in the study

Figure 5.1 shows the change in ACC according to the threshold where it is possible to see that after the initial decrease, ACC increases again until it reaches a peak and it decreases again. The optimal thresholds, corresponding to these peaks, are summarised in Table 5.1 along with some network metrics, describing characteristics of these networks. All examples had a ACC larger than 0.6 and modularity value larger than 0.5, indicating that the resulting networks were indeed modular. The networks were sparse as indicated by the edge density of around 10%.

Table 5.1 Optimal threshold values and network metrics of QSAR data sets

Data set	n	t_α	ACC	Modularity	No. of modules	Edge density	Average degree	Average shortest path	No. of singletons	Degree assortativity
NPYR1	363	0.24	0.64	0.80	98	0.06	20.19	1.54	76	0.74
NPYR2	377	0.24	0.70	0.63	81	0.12	43.64	3.12	64	0.90
CHRM3	643	0.25	0.75	0.56	53	0.10	61.33	3.18	33	0.67
hDHFR	560	0.29	0.80	0.66	20	0.10	53.41	2.89	7	0.69
rDHFR	883	0.29	0.72	0.71	8	0.09	75.44	2.56	1	0.70

A large number of modules $|M|$ are detected by Louvain algorithm at the optimal threshold level in some of these networks. This is due to the occurrence of singletons, molecules that are not similar to any other at the established threshold

level. Figure 5.2 shows that the number of singletons grows with the threshold differently for each data set.

At a $t_\alpha = 0.30$ level, almost a third of the molecules in NPYR1 and NPYR2 are singletons while hDHFR and rDHFR are the most homogeneous data sets in this study, contain less than 10 of these isolated modules. Despite the presence of outliers, there are usually 10 or less well-defined modules in these data sets, as shown in Figure 5.3.

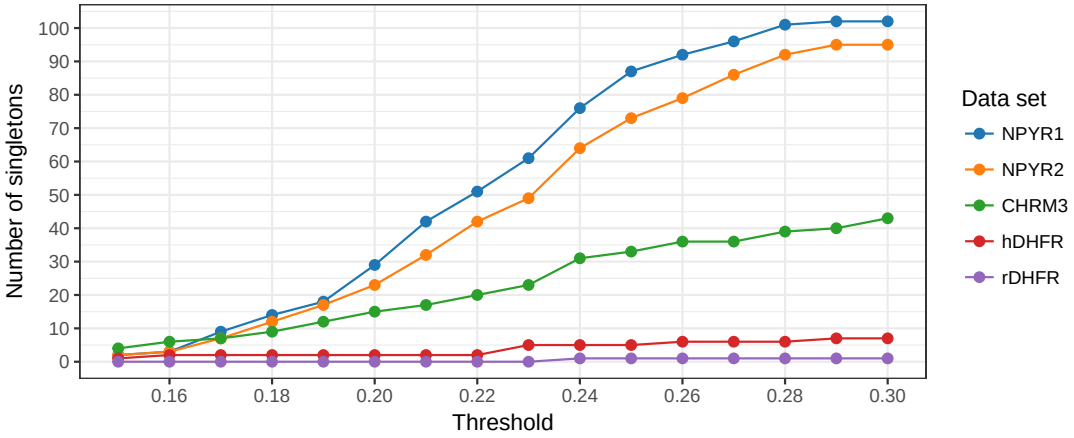
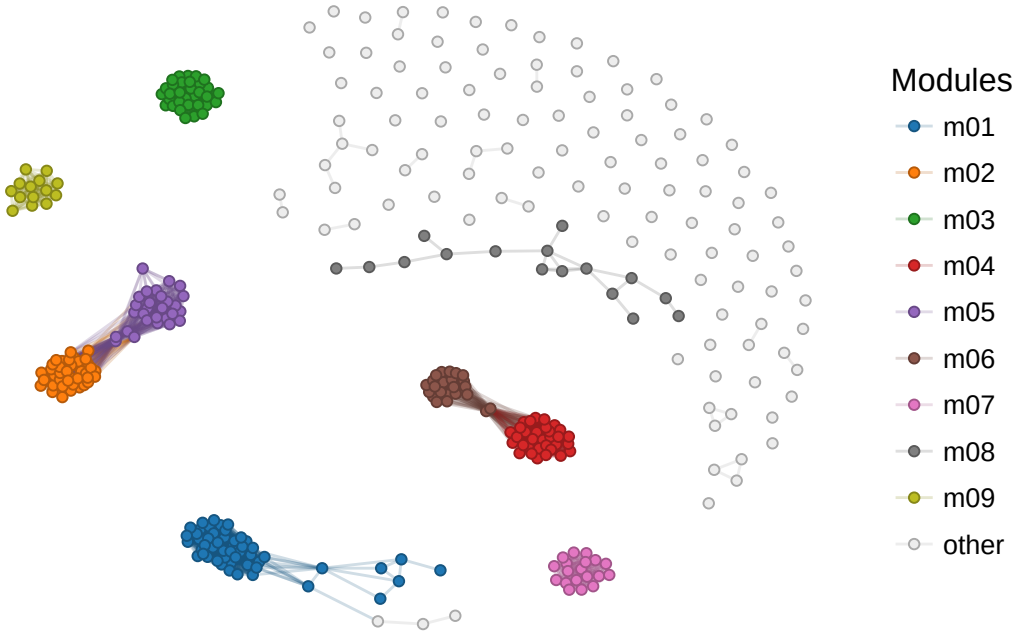


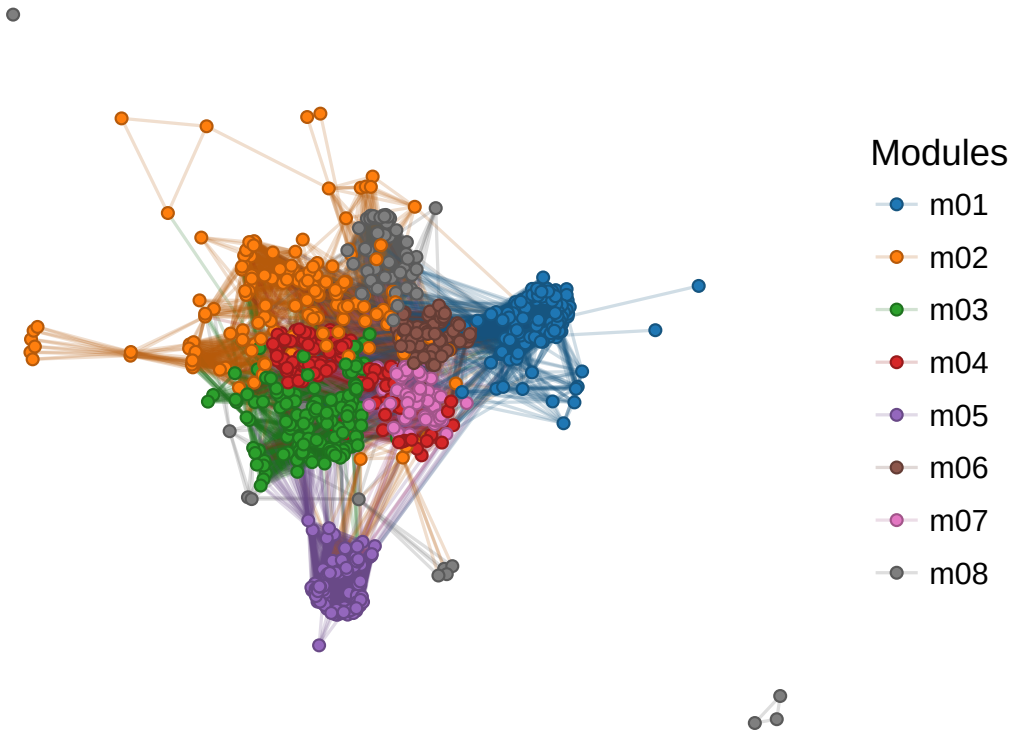
Fig. 5.2 Number of singletons for each data set according to threshold level in the interval $0.15 \leq t_\alpha \leq 0.30$

5.2.2 Presence of activity cliffs

Activity cliffs are discontinuities in structure-activity relationships where molecules with similar structures have a large variation in activity response. Activity cliffs can be measured numerically and molecules classified as "high", "intermediate" or "low" depending on its activity discontinuity [158]. If a molecule is labelled "high" in activity cliff (AC), most of its neighbours, although structurally similar, have an unexpected difference in biological activity; if low, the variance in activity in the neighbourhood of a molecule is more easily explained by their dissimilarities. The proportion of activity cliff classes in these networks is large. Table 5.2 shows that



(a) NPYR1 network ($t_\alpha = 0.24$)



(b) rDHFR network ($t_\alpha = 0.29$)

Fig. 5.3 Examples of network visualisations generated with optimal thresholds

more than half of the samples in all examples are classified as either intermediate or high.

Table 5.2 Proportion of activity cliff classes in the QSAR data sets studied.

Dataset	Discontinuity Class		
	High (%)	Interm. (%)	Low (%)
NPYR1	22.65	35.54	41.81
NPYR2	21.73	33.23	45.05
CHRM3	21.15	30.66	48.20
rDHFR	21.32	34.24	44.44
hDHFR	25.86	28.39	45.75

Activity cliffs are not uniformly distributed in the network. Figure 5.4 shows the network of NPYR1 inhibitors where nodes were coloured according to their activity classes. Notice that some modules consist entirely of samples high in AC (module m04, as compared to Figure 5.3a), while other modules consist in a mix of low, intermediate and high classes. The impact in IC50 predictions of this unequal AC proportion in modules is discussed with more details in the Results section.

5.2.3 Structural properties of modules

Data in chemical libraries are not generated at random but the records evolve over time according to design efforts in the research community [233]. Usually, an initial pool of interesting chemical compounds are identified after a initial high throughput screening (HTS) and are further examined. The chemical space is vast and it is improbable that new dissimilar compounds will be tested against the drug target if promising scaffolds were already identified. Instead, researchers usually optimise these promising compounds by proposing small structural alterations until a molecule with the desired properties is found. As a consequence of this

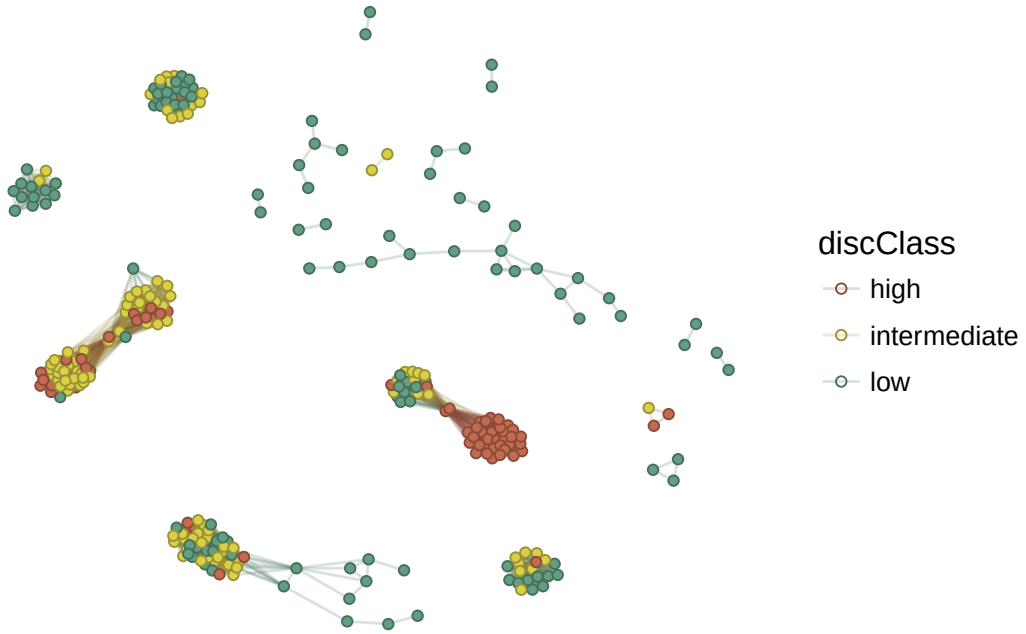


Fig. 5.4 Activity cliffs in NPYR1 network

process, molecules that belong to the same module in a molecular network are also likely to have been recorded in the same bioassay.

One example is module m05 of rDHFR network represented in Figure 5.3b. All samples of this module were obtained from a specific group of assays and no other compound from these assays were present in any other module in the network. The references in the data reveal a long running series of papers [234–239] published by a common group of researchers about the structural studies of 2,4-diamino-5-aryl-6-ethylpyrimidines derivatives as dihydrofolate reductase inhibitors (Figure 5.5).

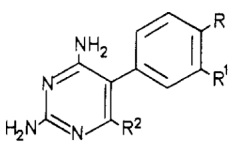


Fig. 5.5 Common structure of molecules in module m05 of rDHFR network. Source: [234]

The module-assay correspondence of all modules of rDHFR network can be visualised in Figure 5.6. The stripes of the diagram illustrate the proportion of samples in each module that are part of a specific or mixed group of assays. Note that samples of module m05 were gathered from a unique group of assays, as described above. In fact, the majority of samples in all modules except m06 came from a unique groups of assays and only modules m04 and m06 were more diverse and were not associated with any particular group of experiments.

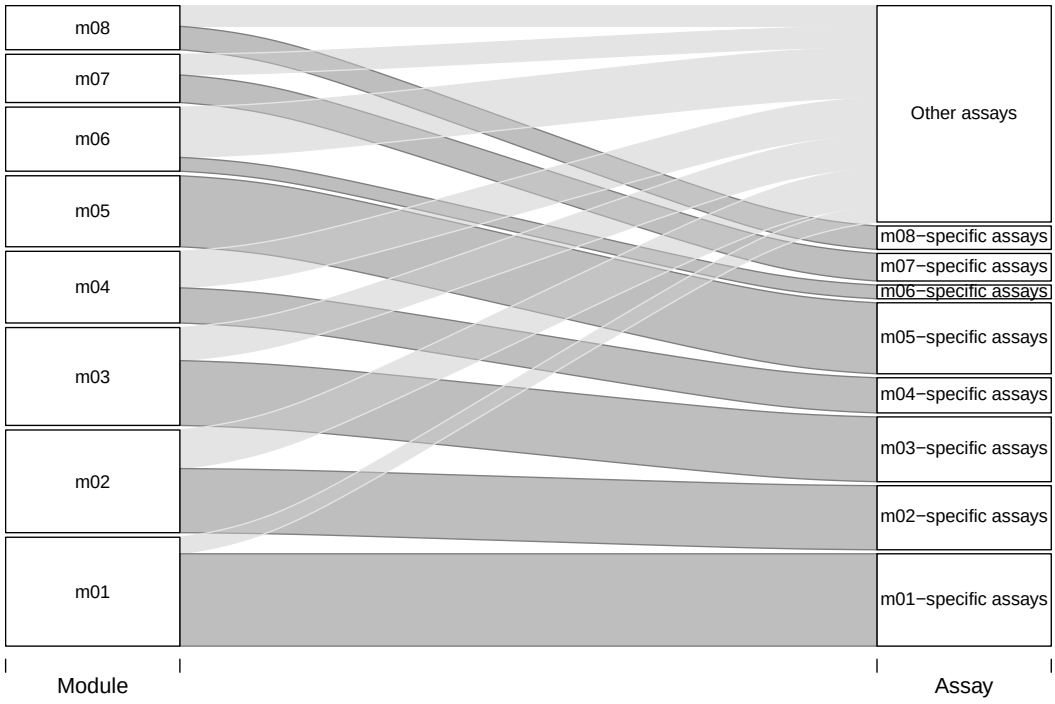


Fig. 5.6 Module-assay correspondence for network rDHFR

These modules can also be characterised by their common structural core. Figure 5.7 shows the maximum common substructure (MCS) of samples in rDHFR modules computed using RDKit [139]. Notice that the MCS captured for samples in module m05 is even more specific that the core identified in the literature (Figure 5.5). Some modules have distinct substructures (m04, m05 and m06); others only have a ring in common (m02 and m03) and some modules are described by more generic fragments (m01 and m08). In these cases, the Murcko scaffold

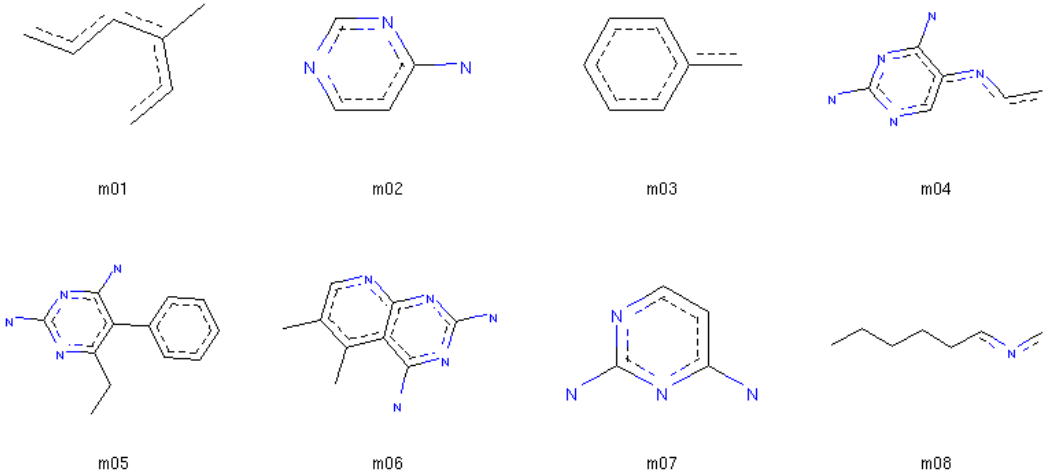
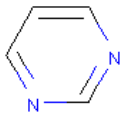


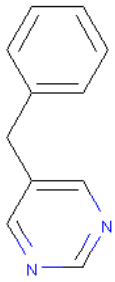
Fig. 5.7 Maximum common structure of each module in rDHFR network

[240] can alternatively provide a description of the most common fragments in a module. This method splits molecules into rings systems, linkers (atoms linking rings), side chains and frameworks (union of rings and linkers) and is a useful tool to identify common graph motifs in a set of compounds, particularly for groups with generic MCS. Diverse modules such as m01 and m08 of rDHFR network, for example, could be better characterised by the rings and frameworks shown in Figure 5.8.

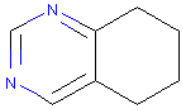
It is not just large modules that are associated with assays; singletons also provide interesting information about the source of the data. The neuropeptide receptor inhibitors data sets (NPYR1 and NPYR2) have the largest number of singletons among the data sets studied and nearly all singleton modules in these networks are associated with a group of PubChem records [241–243] as part of the same high throughput screening assay project entitled "Summary of probe development efforts to identify antagonists of neuropeptide Y receptor Y2 (NPY-Y2)" [244]. Some of these compounds were later explored in other SAR studies and are represented in larger modules such as m01 and m02 of Figure 5.3a.



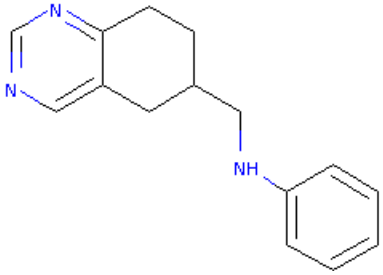
(a) Most common ring in m01 (99% of samples)



(b) Most common framework in m01 (68% of samples)



(c) Most common ring in m08 (64% of samples)



(d) Most common framework in m08 (27% of samples)

Fig. 5.8 Most common rings and frameworks for modules m01 and m08 in rDHFR network

5.3 Network algorithm

Here a new method, Modular (OPLRA), is presented to build interpretable QSAR sub-models from modules in network representations of QSAR data sets. Different to other studies, where clustering is used to separate the data before the model is built [108] or to perform analysis on activity cliffs, module detection is an integral part of our algorithm and are used directly for prediction of biological activity. Modular (OPLRA) perform a hierarchical division of the data where the first layer contains modules detected from molecular networks which are then subdivided into regions identified by the algorithm introduced in the previous chapter, OPLRAreg.

A summary of the proposed method is represented in Figure 5.9. First, a similarity matrix between every pair i and j of molecules is calculated using the

ECFP4 fingerprints (FP) and Tanimoto coefficient (Tc), and the optimal threshold (t_α) is applied to obtain an adjacency matrix representing and unweighted network. Then, the Louvain algorithm [57] identifies modules in the network M that are then modelled independently with OPLRAreg. Each group contain its own regression coefficients and might be separated in one or more regions, as defined by the breakpoints in OPLRAreg (Chapter 4). The Louvain algorithm is a fast algorithm and produces high quality solutions and therefore, it was chosen as the community detection method over other mathematical programming approaches as a more suitable algorithm for this computationally expensive cross-validation workflow.

The output of the algorithm is a graph g where nodes, samples in the trained graph s_{train}, are labelled according to their module membership m and each module is effectively a OPLRA regularised model, identified as OPLRA$_m$, containing its own breakpoints, regions and regression coefficients.

Depending on the diversity of the chemical space present in the data set and the network representation, there will be a number of singleton modules. Singletons can sometimes be considered structural outliers and removed from QSAR models [108] since these molecules are not well represented in the data sets. We have nonetheless opted to keep singletons in our analysis since, in our proposed algorithm, these compounds do not interfere with the predictions of other more populated modules. Because it is not necessary to run OPLRA for a single compound, singleton modules in the training step are fit by the actual value of its experimental pIC_{50}.

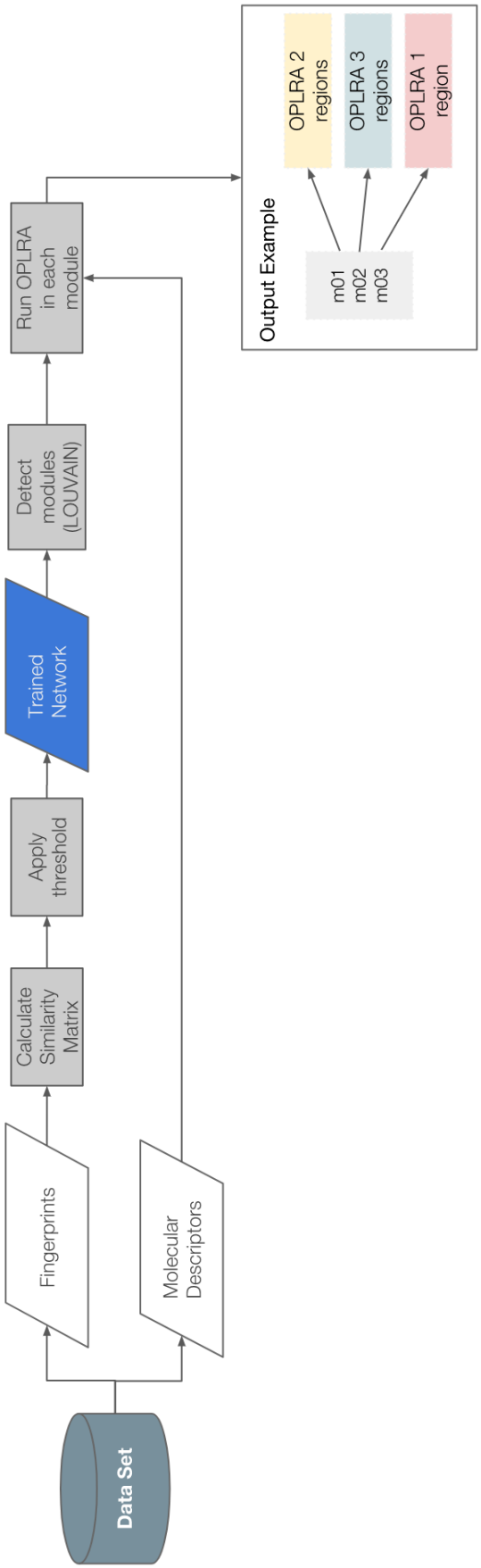


Fig. 5.9 Modular (OPLRA) algorithm

5.3.1 Prediction of new samples

To determine the neighbourhood of a test sample s_{test}, we calculate the similarity of s_{test} to all samples in the trained graph. The module of s_{test} is then determined according to one of the possible three cases below:

Case 1 Test sample has many connections to trained data. In the simplest case, s_{test} is assigned to the module to which it has more connections above the threshold t_α.

Case 2: Test sample is equally connected to multiple modules. When the test sample has a maximum number of connections to more than one module, we assign it to the module corresponding to the largest average similarity.

Case 3: Test sample does not have any neighbours in the graph. In this case, the assign s_{test} to the module of its nearest neighbour in the graph. Note, however, that these predictions are not reliable, since s_{test} is dissimilar to samples in the trained data and it can be considered to be outside the applicability domain (AD) of the QSAR model. We flag such predictions as unreliable and we remove it from the statistics of the models.

5.3.2 Implementation details and algorithm validation

Molecular descriptors and ECFP4 fingerprints were computed using Chemistry Development Kit (CDK) [138] implemented in the R package rcdk [203]. The algorithm and the cross-validation procedure was implemented in Python while mathematical programming models in OPLRAreg were developed with Pyomo version 5.2 and solved with CPLEX MIP solver. The validation procedure is the same as the presented in Figure 4.2. We ran 5 batches of tests, where we split

the data at random into internal (75%) and external (25%) sets. For each round, we ran 10 rounds of 10-fold cross validation and selected the model with the best performance in the internal set to predict samples in the external set. The regularisation parameter was selected using grid search inside the cross-validation for the possible values: $\lambda \in \{0.005, 0.05, 0.10\}$.

5.4 Results and Discussion

In this section, the QSAR models produced by Modular (OPLRA) algorithm are presented and discussed for each group of data set. A description of the overall performance in internal, validation and external sets of the examples is presented, followed by a discussion about the robustness of modules found during cross-validation. Examples of results given by the algorithm are shown next. Networks, piecewise models and predictions per modules are presented, along with a discussion about singletons and activity cliffs present in the data sets studied. The following section discusses the advantages and disadvantages of Modular (OPLRA) over the original piecewise algorithm and, in the end, a comparative analysis with state of the art machine learning algorithms.

5.4.1 Overall performance

The performance of Modular (OPLRA) is summarised in Tables 5.3 (internal) 5.4 (external validation), where mean absolute error (MAE) and standard deviation (SD) of prediction errors are represented for each of the five batches of experiments. In the internal training and validation, Modular (OPLRA) had a similar result independent of the data split. The average error in the validation set for NPYR1, NPYR2 and rDHFR data sets was $MAE \approx 0.60$, but NPYR1 and NPYR2 had an

Table 5.3 Performance of Modular (OPLRA): average mean absolute error and deviation in the cross-validation

Data Split	Data Set				
	NPYR1	NPYR2	CHRM3	hDHFR	rDHFR
Training					
1	0.27 (±0.15)	0.26 (±0.13)	0.48 (±0.10)	0.39 (±0.17)	0.46 (±0.14)
2	0.19 (±0.10)	0.26 (±0.12)	0.46 (±0.13)	0.44 (±0.17)	0.38 (±0.12)
3	0.24 (±0.10)	0.26 (±0.13)	0.44 (±0.13)	0.39 (±0.17)	0.44 (±0.14)
4	0.24 (±0.10)	0.25 (±0.13)	0.44 (±0.13)	0.49 (±0.19)	0.35 (±0.12)
5	0.24 (±0.12)	0.26 (±0.13)	0.50 (±0.12)	0.42 (±0.19)	0.36 (±0.11)
Validation					
1	0.63 (±0.15)	0.58 (±0.09)	0.71 (±0.09)	0.68 (±0.09)	0.63 (±0.08)
2	0.57 (±0.12)	0.56 (±0.10)	0.75 (±0.09)	0.72 (±0.08)	0.59 (±0.07)
3	0.60 (±0.14)	0.57 (±0.10)	0.68 (±0.09)	0.71 (±0.10)	0.62 (±0.07)
4	0.60 (±0.14)	0.57 (±0.10)	0.74 (±0.09)	0.76 (±0.09)	0.58 (±0.07)
5	0.63 (±0.14)	0.58 (±0.10)	0.70 (±0.08)	0.72 (±0.09)	0.58 (±0.07)

Table 5.4 Average performance of Modular (OPLRA) in the external set

Data Split	Data Set				
	NPYR1	NPYR2	CHRM3	hDHFR	rDHFR
Inside AD					
1	0.53 (±0.46)	0.57 (±0.51)	0.67 (±0.52)	0.67 (±0.58)	0.54 (±0.49)
2	0.61 (±0.63)	0.49 (±0.50)	0.60 (±0.55)	0.62 (±0.61)	0.56 (±0.53)
3	0.62 (±0.65)	0.53 (±0.44)	0.72 (±0.54)	0.68 (±0.75)	0.59 (±0.57)
4	0.58 (±0.54)	0.59 (±0.70)	0.65 (±0.53)	0.64 (±0.67)	0.61 (±0.62)
5	0.67 (±0.57)	0.59 (±0.50)	0.68 (±0.55)	0.71 (±0.77)	0.55 (±0.54)
Outside AD					
1	0.44 (±0.61)	0.46 (±0.55)	1.23 (±0.89)	0.50 (±0.00)	-
2	0.30 (±0.79)	0.42 (±0.38)	1.70 (±1.12)	2.20 (±2.29)	-
3	0.57 (±0.70)	0.37 (±0.30)	1.63 (±1.19)	-	-
4	0.73 (±1.23)	0.38 (±0.41)	1.93 (±1.35)	0.45 (±0.00)	0.69 (±0.47)
5	0.45 (±0.57)	0.56 (±0.61)	1.32 (±1.16)	1.10 (±0.92)	1.88 (±0.00)

even smaller average MAE in the training samples ($MAE \approx 0.25$), suggesting these datasets were easier to model. Modular (OPLRA) also predicted the validation set of rDHFR with $MAE \approx 0.60$ but the error in the training samples was slightly larger ($MAE \approx 0.40$).

The error of predictions in the external set also revolve around 0.60 but the standard deviation is larger, $SD \approx 0.50$ and $SD \approx 0.70$ in some cases. This large error variation indicates the difficulty of modelling heterogeneous QSAR data sets and it is not unique to Modular (OPLRA); Random Forest and Support Vector Machines also predict these samples with the same margin of error, as it is demonstrated in Section 5.4.8.

Table 5.4 also shows the different predictions made for samples inside and outside the applicability domain (AD). As indicated in section 5.3.1, the applicability domain of Modular (OPLRA) was defined by the similarity of the predicted molecules to those in the graph. When the molecule to be predicted is not similar to any other molecules in the graph, above the threshold, we say it is out of the applicability domain and the prediction in this case is not very reliable. Note that the algorithm has produced more erroneous predictions for samples outside AD in CHRM3, hDHFR and rDHFR data sets, the predictions were wrong by 1 or 2 log units on average. However, NPYR1 and NPYR2 have produced similar and even better average predictions for samples outside AD in some data splits.

The large error variation can be explained in part by the presence of activity cliffs in the modules. The correlation between the proportion of activity cliffs and the range of prediction errors produced by each module was investigated and it is shown in Figure 5.10. Dots in these plots represent all modules identified during all five training rounds, the x-axis shows the proportion of intermediate and high activity cliffs present in the modules so that the leftmost points indicate modules with fewer activity cliffs, the y-axis represents error variation by the interquartile range of absolute errors of predictions made in the external set. NPYR1 was the example were these two values were most correlated (0.63), modules with a higher proportion of activity cliffs were most likely to have a larger error dispersion. The correlation is relatively large for most data sets, except for hDHFR (cor = 0.29).

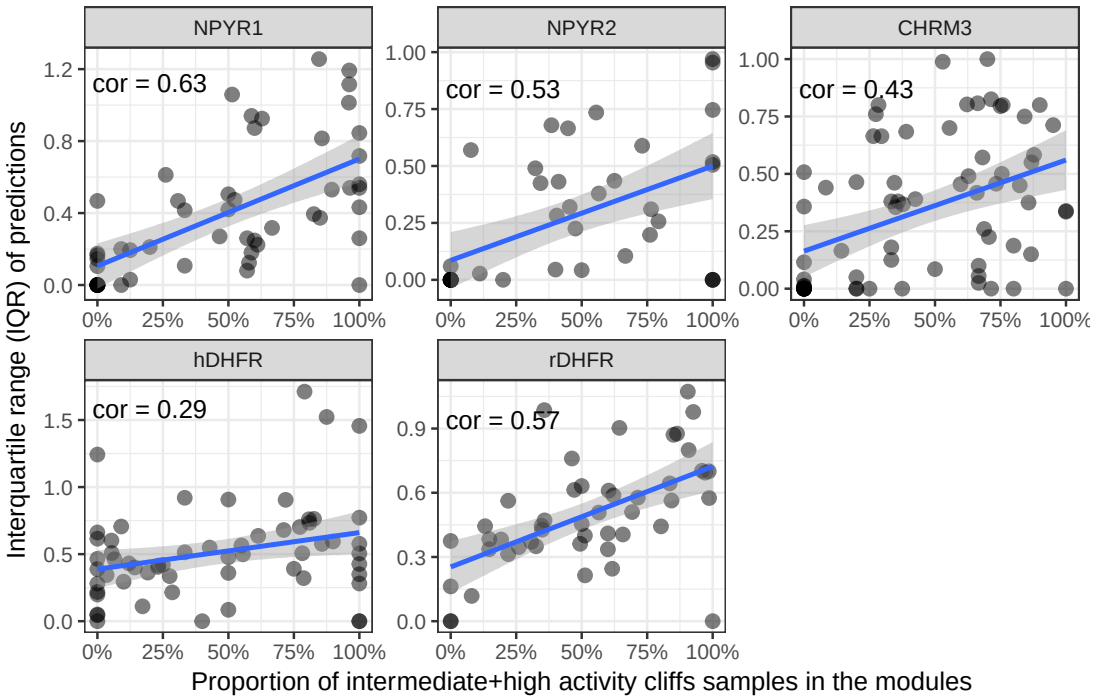
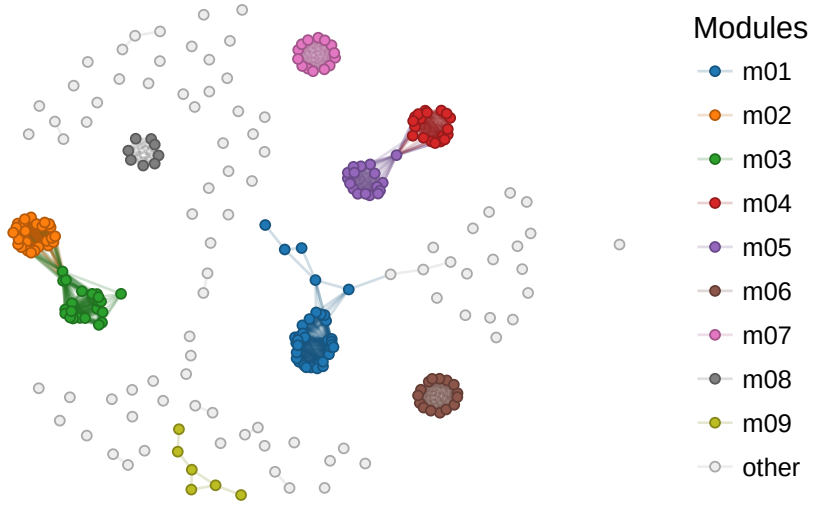


Fig. 5.10 Low activity cliffs and interquartile range of prediction errors in the external set

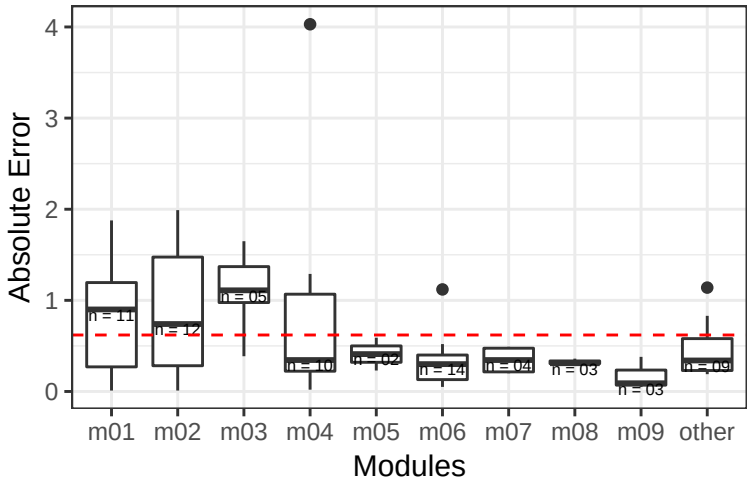
5.4.2 Neuropeptide Y inhibitors

NPY networks in this study are characterised by a large number of singletons and a few well-defined modules disconnected from the rest of the network. In this section, the similarities between the QSAR models of these data sets and the implications of these properties in the predictive modelling are discussed.

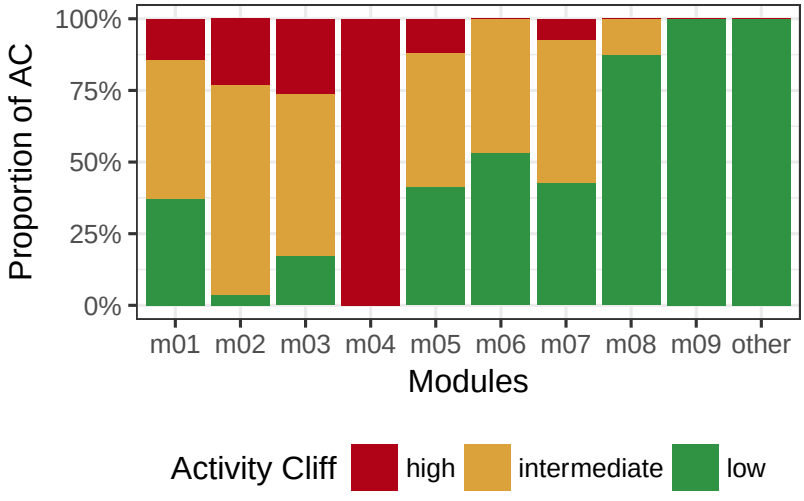
Figures 5.11 and 5.12 show networks obtained during training of Modular (OPLRA), predictions made by each modules in the external set and the proportion of high activity cliffs for NPYR1 and NPYR2 data sets, respectively. NPYR1 network contains 8 main modules while NPYR2 contains 5; both have a large number of small groups and singletons. Box plots show the predictive performance of each module in the examples and the dashed line indicates the mean absolute error of all predictions in the external set combined.



(a) NPYR1 network obtained from training samples



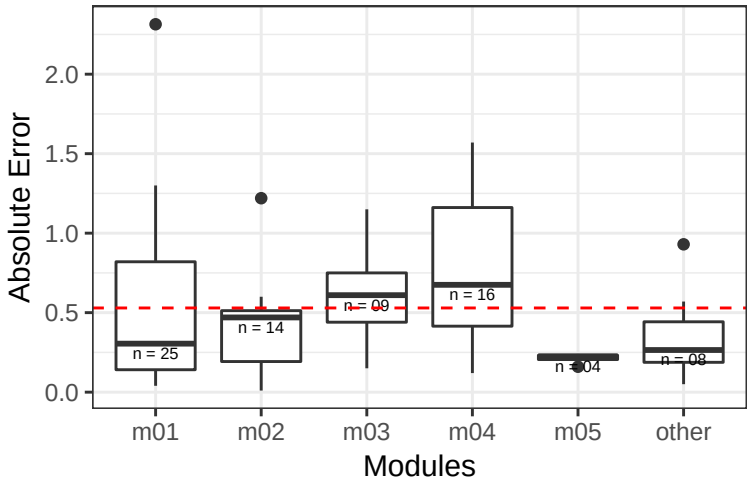
(b) Performance of predictions in the external set



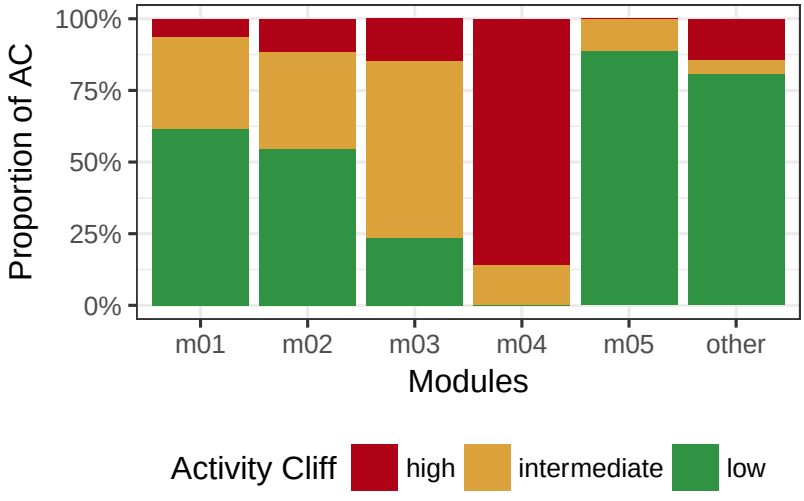
(c) Proportion of activity cliff classes per module

Fig. 5.11 NPYR1 network modules and predictive performance in the external set

(a) NPYR2 network obtained from training samples



(b) Performance of predictions in the external set



(c) Proportion of activity cliff classes per module

Fig. 5.12 NPYR2 network modules and predictive performance in the external set

An anticipated challenge in modelling these data sets was the large number of singletons and small disconnected modules. The chemical space around these molecules is under represented and one would expect that predictions made by these clusters would not be accurate. However, these predictions, shown as "other" in Figures 5.11b and 5.12b, were below the MAE line and only contained one outlier in each case. The dispersion of error was also small, as shown by the interquartile range of the box plot. Larger and well-defined modules, on the other hand such as m01 and m02 in NPYR1 and m01 and m04 in NPYR2 showed a large error variation.

One possible explanation for the larger error dispersion on these modules is the presence of activity cliffs. In fact, some of the modules that predicted large dispersion of errors or large mean absolute error in the external set contained a small proportion of samples free from activity cliffs. For example, compared to the rest of the network, modules m01, m02, m03 and m04 of NPYR1 (Figure 5.11c) and modules m03 and m04 in NPYR2 (Figure 5.12c) have the smaller proportion of low AC samples and correspond to predictions with the larger error dispersion or larger median error. However, this might not be the only explanation for the error variation. Module m05 in NPYR1, for example, has a similar proportion of activity cliffs to module m01 but prediction errors for m05 are much smaller; also, a small error variation would be expected from module m01 in NPYR2 since it consists mainly of low AC samples.

5.4.3 CHRM3

The results for data set CHRM3 are shown in Figure 5.13. CHRM3 network also has a large number of isolated modules but it is more connected than NPY networks, its 8 main modules can be found on its giant component. There was no

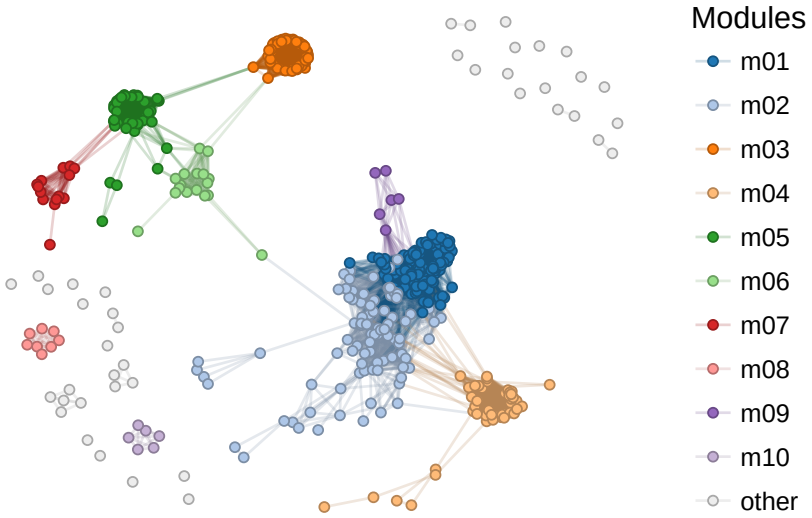
clear relationship in this particular example between the proportion of activity cliffs and the accuracy of predictions but once again modules that have a small proportion of low AC and had a large error variation, e.g. m02, m05 and m07.

In most tests, the optimal value for the regularisation parameter of Modular (OPLRA) was $\lambda = 0.05$ but in this particular example, the selected parameter was $\lambda = 0.10$. A large λ can result in piecewise models that do not contain any regression coefficients in the equations and are defined only by breakpoints, the partition feature and the intercept of regression. As an example, module m06 in Figure 5.13 was separated into 2 regions by OPLRAreg according to the rule below:

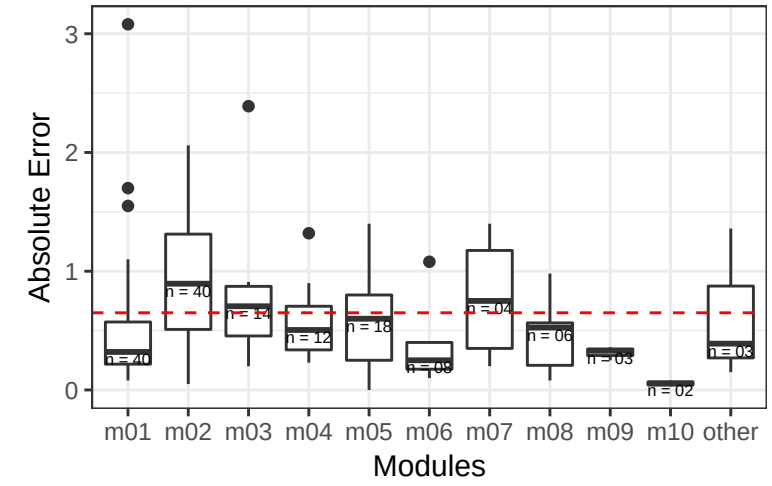
$$\text{pIC50} = \begin{cases} 6.60, & \text{if BCUTw.1l} \leq 0.528 \qquad \text{(region 1)} \\ 8.00, & \text{if BCUTw.1l} > 0.528 \qquad \text{(region 2)} \end{cases}$$

Even though this simple rule does not use any molecular descriptors in the equations, it identifies the BCUT descriptor as an important partition feature to separate compounds in the micromolar range (region 1) from more potent compounds in the nanomolar range (region 2). This piecewise equation also predicts 8 of the 9 samples in the external set with a small error variation (0.24 ± 0.13), the only outlier is in region 1 where a sample had an experimental pIC50= 5.52, instead of the predicted 6.60.

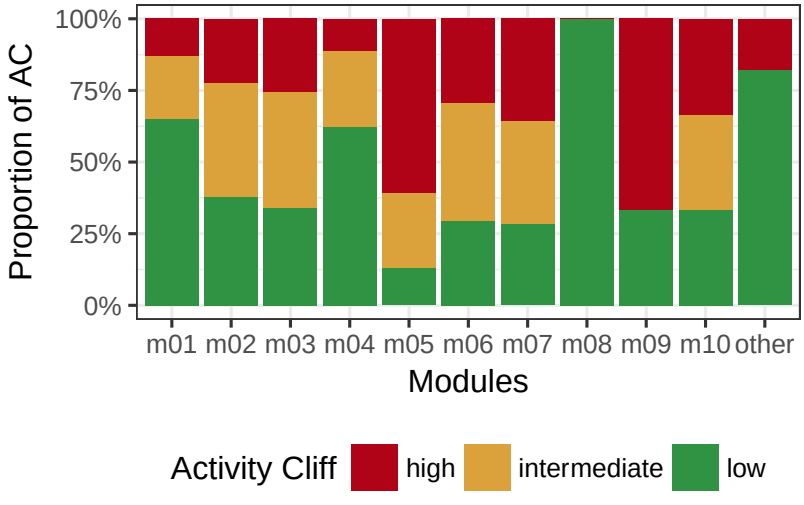
Although BCUT descriptors have been shown to make significant contributions to QSAR models [245], these features are not very intuitive. BCUTw.1l, for example, represents the first eigenvalue of a matrix representation of a molecule where the diagonals contain atomic weights. A more intuitive form to characterise groups of molecules is to identify their common structural core. For this particular



(a) CHRM3 network obtained from training samples



(b) Performance of predictions in the external set



(c) Proportion of activity cliff classes per module

Fig. 5.13 CHRM3 network modules and predictive performance in the external set

example, the maximum common substructure (MCS) of samples in module 06 were computed using RDKit [139] and are shown in Figure 5.14. Notice that molecules in Region 2 (Figure 5.14b) have a more extensive structural core than the MCS identified for samples in region 1 (Figure 5.14a). This information, along with the BCUTw.1l descriptor value could help devise new active compounds in this series.

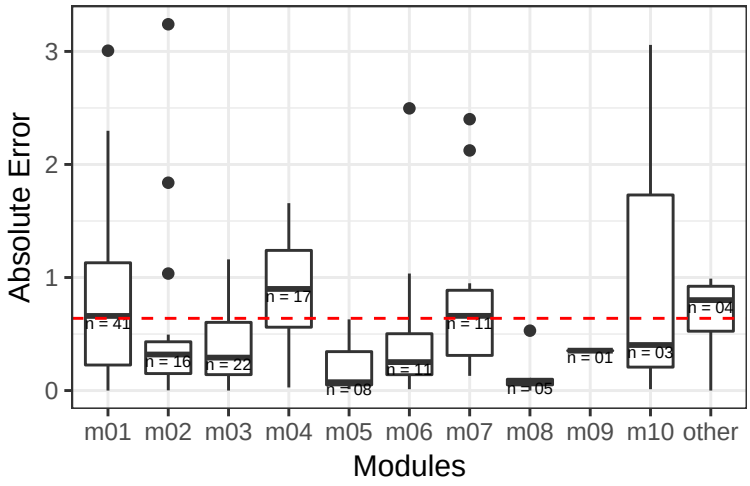
(a) MCS of samples in m06 (region 1) (b) MCS of samples in m06 (region 2)

Fig. 5.14 Maximum common substructures of samples m06 in CHRM3 network

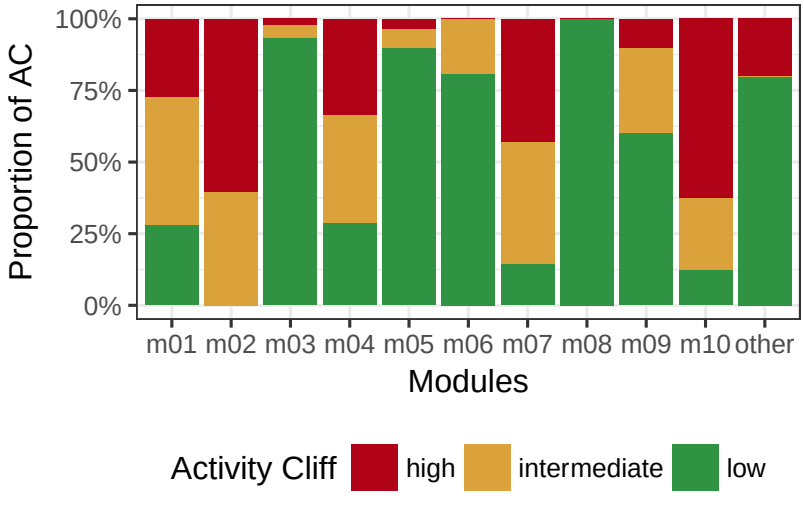
5.4.4 DHFR inhibitors

The modules and predictions made by Modular (OPLRA) in the data sets of rat and human DHFR inhibitors are shown in Figures 5.16 and 5.15. These networks contain fewer singletons than the other data sets and the modules are denser and more interconnected. As a consequence, few samples in the external set are outside the applicability domain and most samples can be predicted by Modular (OPLRA) sub-models created during cross-validation (Table 5.3). On average, the performance of Modular (OPLRA) in rDHFR is similar to the other data sets, the mean absolute error and standard deviation of absolute errors in this data set is (0.57 ± 0.55), averaged across all five data splits. But the error dispersion is relatively larger in hDHFR, where the MAE and SD of prediction errors averaged across all data splits is (0.66 ± 0.68). This large error variation can also be noticed on the box plots of Figure 5.15b.

(a) hDHFR network obtained from training samples

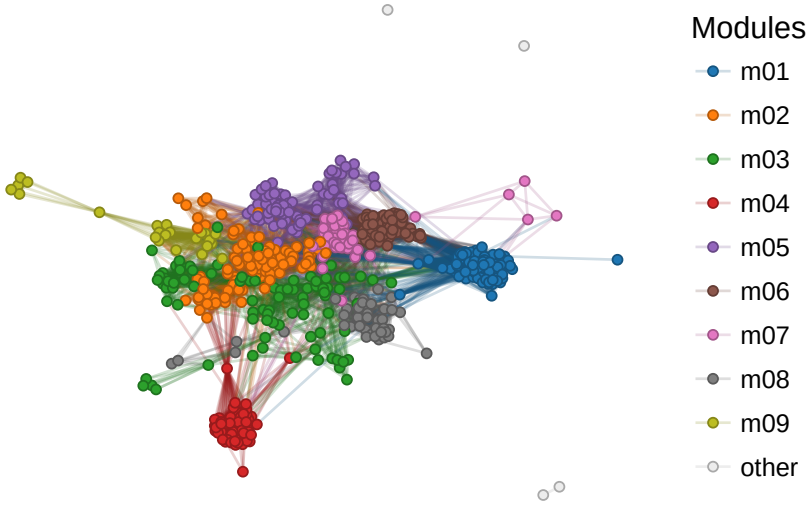


(b) Performance of predictions in the external set

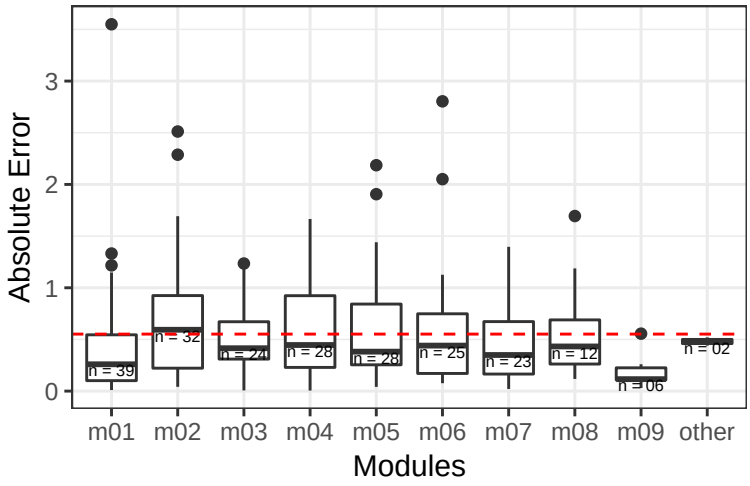


(c) Proportion of activity cliff classes per module

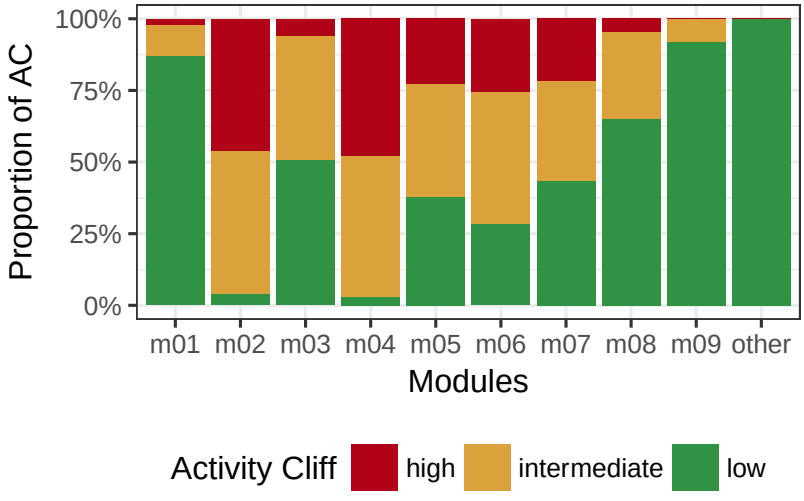
Fig. 5.15 hDHFR network modules and predictive performance in the external set



(a) rDHFR network obtained from training samples



(b) Performance of predictions in the external set



(c) Proportion of activity cliff classes per module

Fig. 5.16 rDHFR network modules and predictive performance in the external set

5.4.5 Robustness of modules

One might question the robustness of the modules found by the algorithm. Because of cross-validation, networks in these tests are created from sub-samples of data and the modules detected in each fold do not entirely match those created from the full data set. For example, in Figure 5.3a, if even one of the nodes in module m07 are not present in the training samples, this module might be split into two; if a large number of molecules are missing, this module might not be present in the trained graph at all. It is easy to see, from this example, that small and sparse modules are the most affected by sub-sampling.

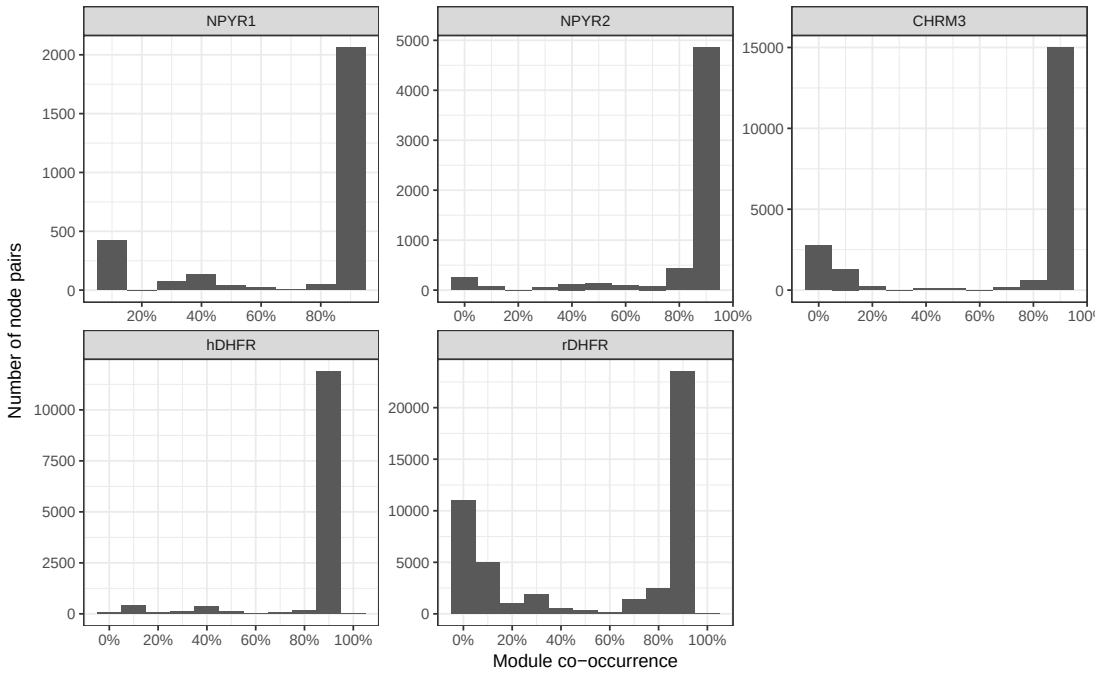


Fig. 5.17 Frequency of co-occurrence of nodes in the same module averaged for 100 sub-sampled networks.

Larger and denser modules, however, are usually well represented independent of the sub-sampling. To confirm this, the 100 networks created during cross-validation for each data set were stored and compared. Then I selected all pairs of nodes that appeared in a same module and counted how many times they co-occurred across folds. The results are presented in Figure 5.17 and show

the frequency that pairs of nodes appear in the same module across all 100 examples. Most pairs of nodes (90+%) appear in the same modules, suggesting that, although networks built during cross-validation have different connectivity than the full graph due to the sub-sampling, the modules are not greatly impacted. Exceptionally, rDHFR data set contain a considerable proportion of samples with a low co-occurrence rate even though most nodes get clustered in the same module, probably because of the density of inter-modules connections. In rDHFR network (Figure 5.3b), the modules are close to each other and when the network is sub-sampled, nodes at the intersection of modules will be separated from other nodes in its original module.

5.4.6 Improvement over piecewise algorithm

Similar to OPLRA, this new proposed algorithm also separates the data set into disjoint groups and create independent models for each group. The regions found by Modular (OPLRA), however, are more distinctive and informative. Take rDHFR network in Figure 5.16a again as example, where the common core of each modules can be well characterised (Figures 5.7 and Figure 5.8). OPLRAreg separates this same network into two regions (Figure 5.18) with the partition feature MDEN.12. Samples with a smaller molecular distance between primary and secondary nitrogen atoms are included in Region 1, otherwise, they are grouped in Region 2. Note that, because this separation of samples is more generic, dissimilar molecular structures can be assigned to the same module. As a consequence, the common cores of regions will not be as representative of the real clusters in this data set (Figure 5.19).

Modular (OPLRA) also makes better predictions than OPLRA regularised, with a smaller error dispersion. Table 5.5 shows the average reduction in mean

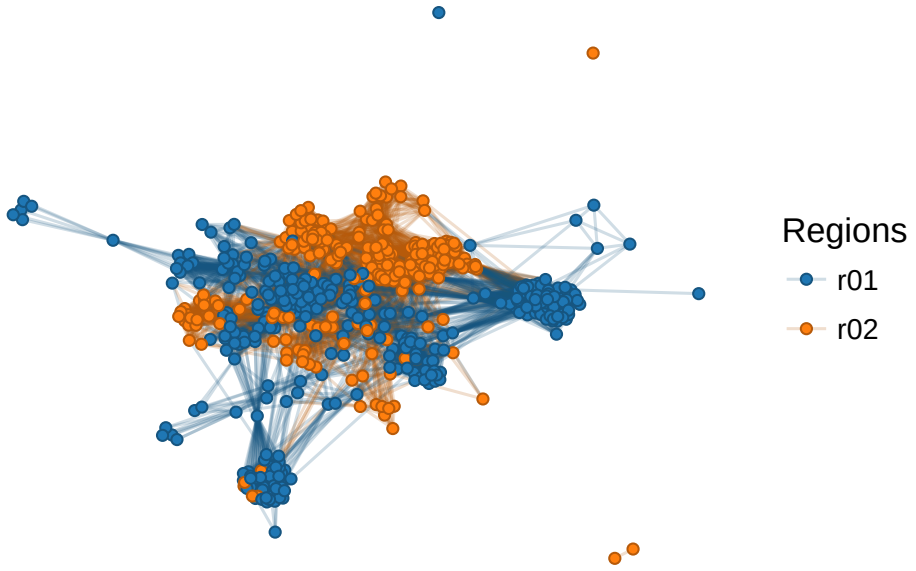


Fig. 5.18 Regions identified by OPLRAreg for dataset rDHFR

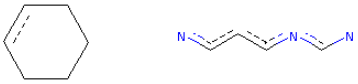


Fig. 5.19 Maximum common substructures of r01 and r02

absolute error (MAE) and in the standard deviation (SD) of absolute errors (shown inside brackets) for external sets predictions. MAE was smaller for samples in intermediate and low discontinuity class in almost all data sets. MAE is also 33%, 24% and 10% smaller for high AC samples in NPYR1, hDHFR and rDHFR data sets, respectively. Modular (OPLRA) shows improvement in SD in all data sets, which means that the dispersion of errors of all cases was reduced, even those with a modest MAE reduction. Notably, CHRM3 was the only data set for which Modular (OPLRA) made a worse prediction for high AC samples, with 6.40% average increase in MAE but the error dispersion was improved considerably, with a reduction of 42.64% in SD.

Table 5.5 Reduction in MAE and SD of errors by Modular (OPLRA) compared to OPLRAreg per discontinuity class

Dataset	Discontinuity Class		
	High (%)	Interm. (%)	Low (%)
NPYR1	-32.87 (-44.88)	-8.06 (-32.44)	-14.44 (-8.61)
NPYR2	0.32 (-14.04)	-1.61 (-25.81)	-8.57 (-6.62)
CHRM3	6.40 (-42.64)	-14.53 (-46.56)	-14.72 (-26.57)
hDHFR	-23.70 (-35.19)	-0.08 (-3.71)	0.12 (-3.93)
rDHFR	-10.23 (-38.30)	-8.31 (-26.62)	-33.28 (-44.10)

5.4.7 Generating constraints for *de novo* molecular design

Molecular *de novo* design is a computational and optimisation technique to generate new chemical entities [246–248]. *De novo* algorithms, usually evolutionary algorithms, combine fragments and functional groups following certain imposed constraints and propose structures predicted to give improved biological activity [249, 250].

Any predictive and validated QSAR model can be used to score the activity of these artificial compounds. But Modular (OPLRA) could have a more important role in *de novo design* as it could be used to generate additional constraints for these algorithms. As an example, suppose we are interested in identifying more potent compounds for NPYR2 and we decide to explore the module m01 shown in 5.12. Modular (OPLRA) has identified the following rules for this module:

$$\text{pIC50} = \begin{cases} -0.252\ \text{MDEC.22} + 7.460, & \text{if khs.aaN} \leq 0.49 \quad \text{(region 1)}, \\ 6.601, & 0.49 < \text{if khs.aaN} \leq 0.74 \quad \text{(region 2)}, \\ 4.610, & \text{if khs.aaN} > 0.74 \quad \text{(region 3)}, \end{cases}$$

where the partition feature khs.aaN represents the number of occurrences of the fragment represented in Figure 5.20 in the molecule. Because the unscaled range of khs.aaN in this data set is $[0, 4]$, samples in region 1 have fewer than 2 occurrences of the fragment, region 2 contains 2 occurrences and region 3 contains 3 or 4.

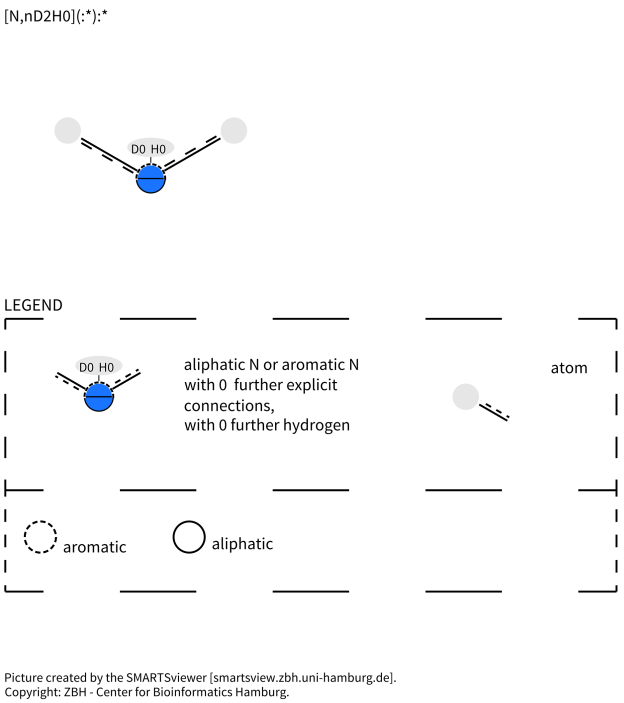


Fig. 5.20 Fragment represented by descriptor khs.aaN

The most promising path to generate a new compound in this series is by producing molecules in region 1 of the OPLRA model above. Samples in this region were predicted to be more potent (minimum pIC50 = 7.460) than the constant numerical predictions of regions 2 and 3. The linear equation of region 1 also showed that activity was negatively correlated with MDEC.22, a descriptor related to the average molecular distance between secondary carbons.

These breakpoints, equations and the neighbourhood of m01 molecules could be combined to provide the following constraints to a *de novo* technique for new NPYR2 inhibitors:

Neighbourhood constraint: new chemical entities should be within the applicability domain of module m01. A new molecule is only considered feasible if its similarity to at least one of the compounds in the module is above the threshold $t_\alpha = 0.24$,

Breakpoint constraint: new molecules must satisfy khs.aaN < 2. There must be a maximum of one nitrogen atom connected to a hydrogen and two different atoms, as represented in Figure 5.20,

Equation constraint: MDEC.22 must be minimised, secondary carbons should be placed next to each other in the molecular graph.

Similar constraints could be obtained for all other modules identified in NPYR2 network and in combination, provide a targeted set of optimisation constraints for *de novo* design. These constraints could potentially reduce the search space of current algorithms, facilitating the generation of synthetically accessible new compounds.

5.4.8 Comparative Analysis

The performance of Modular (OPLRA) was comparable to other machine learning algorithms, as shown in Figure 5.21. As discussed in Section 5.4.6, the mean absolute error and error variation of Modular (OPLRA) – represented by the interquartile range of the box plots – were better than OPLRAreg. The simple piecewise linear algorithm introduced in the previous Chapter 4 already had an average performance comparable to other widely used and predictive algorithms but Modular (OPLRA) represents a further improvement as this method is even more similar to the results of Random Forest and Support Vector Machine. This

highlights once more the suitability of Modular (OPLRA) for QSAR models, as a more interpretable alternative to black-box machine learning algorithms.

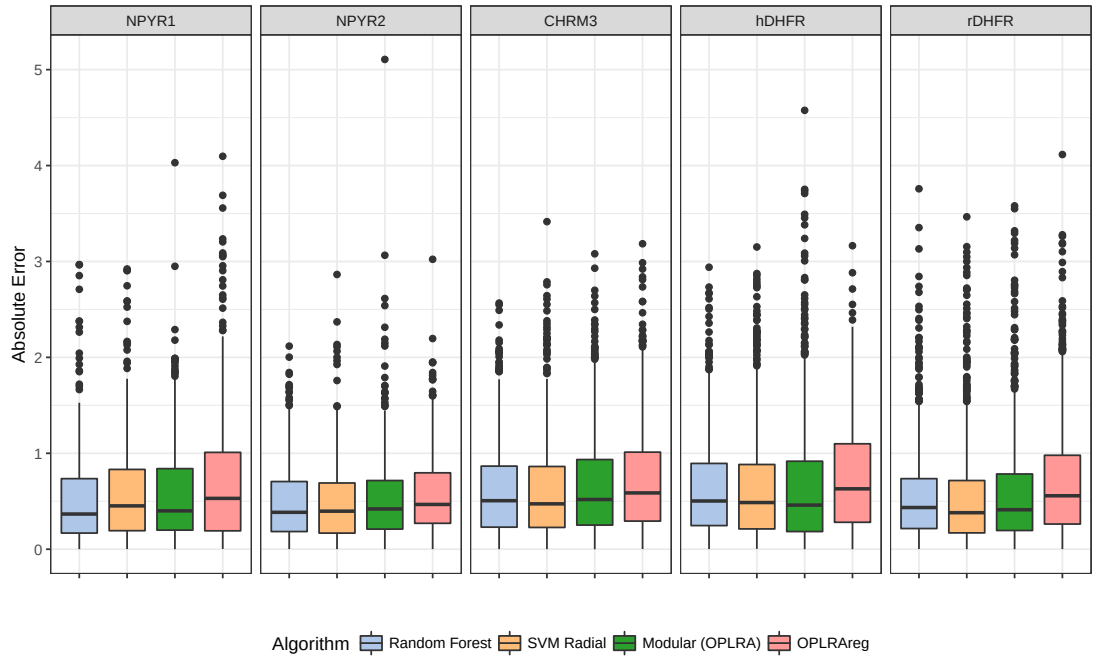


Fig. 5.21 Comparative results of machine learning algorithms in QSAR data sets

5.4.9 Enforcing a minimum number of neighbours

An alternative technique to construct networks that has not been investigated in the QSAR community yet is to enforce a minimum number of k neighbours in the network [198]. If a node i has less than k neighbours under the established threshold level, we connect it to its k-nearest neighbours. Ideally, k should be small so as to not fundamentally change the structure of the network and to put singletons in the appropriate larger modules. Under this new representation, links between molecules become weighted by the Tc similarity, to better represent the strength of the connections.

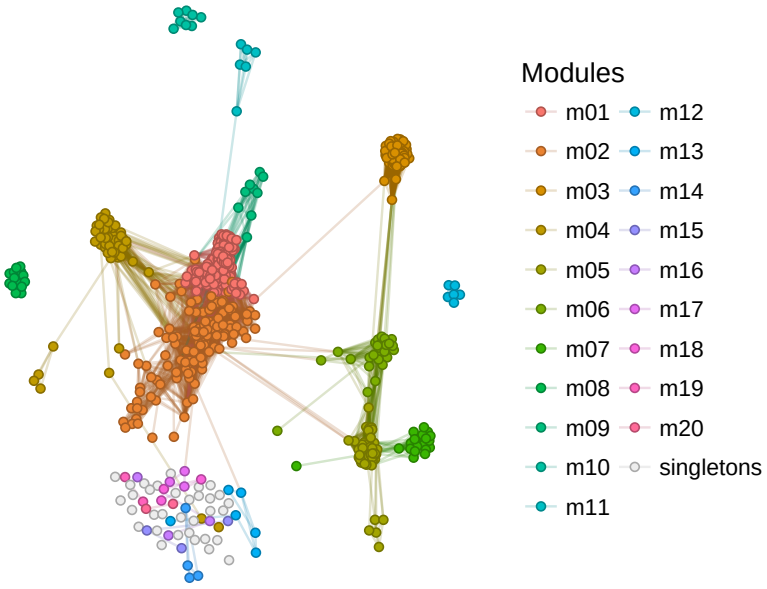
Take CHRM3 data set as an example. Figure 5.22 compares the modules found in the network at the optimal threshold for $k = 0$ (the original network)

and $k = 3$, enforcing a minimum of 3 neighbours. The original network (Figure 5.22a) had 53 modules, 33 singletons and 6 small disconnected modules shown at the bottom of the graph, while the network built with $k = 3$ (Figure 5.22a) only had 14 modules. All singletons and those small modules with less than 4 samples had molecules from module m01 among their k-nearest neighbours and were therefore assigned to m01.

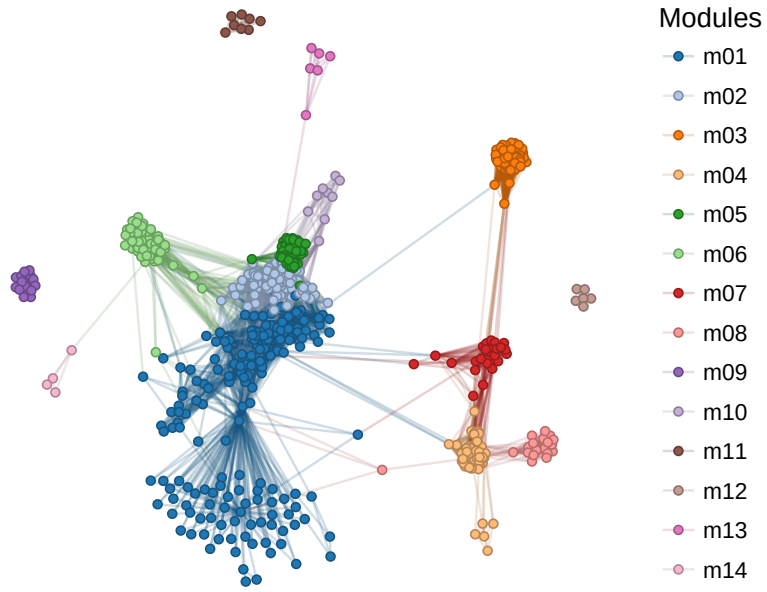
The metrics of networks built with this hybrid construction approach are summarised in Table 5.6. Compared to the original network (Table 5.1), the number of modules decreased considerably in all data sets since all singletons were merged into larger modules. Besides the number of modules, most network properties either remained the same or changed slightly in CHRM3, hDHFR and rDHFR data sets. Properties of the data sets with the larger number of singletons, NPYR1 and NPYR2, showed more significant changes.

NPYR1 network went from assortative in the original network to dissortative in the hybrid construction approach as the degree assortativity changed from 0.74 to -0.34. In the original approach, nodes were connected to nodes of a similar degree but when a minimum of 3 neighbours is enforced, singletons were connected to the rest of the network and high degree nodes were more likely to connect to low degree nodes. The connection of former singleton nodes also changed the degree assortativity of NPYR2 network from 0.90 to 0.23 but NPYR2 still remained assortative. The average shortest path was also affected. The property decreased from 1.54 to 2.75 in NPYR1 and increased from 3.12 to 2.64 in NPYR2, but because singletons were not added to the average calculations in the first network, these changes cannot be compared.

The performance of Modular (OPLRA) using $k = 3$ in the external set are shown in Table 5.7. In comparison to the original algorithm (Table 5.4), the



(a) CHRM3 network
($t_\alpha = 0.25$, $k = 0$, unweighted)



(b) CHRM3 network
($t_\alpha = 0.25$, $k = 3$, weighted)

Fig. 5.22 Examples of network visualisations generated with optimal thresholds

Data set	ACC	Modularity	No. of modules	Edge density	Average degree	Average shortest path	No. of singletons	Degree assortativity
NYPR1	0.93	0.81	9	0.06	22.03	2.75	0	-0.34
NYPR2	0.75	0.67	5	0.12	45.23	2.64	0	0.23
CHRM3	0.78	0.62	14	0.10	61.85	3.11	0	0.66
hDHFR	0.82	0.69	13	0.10	53.54	2.99	0	0.68
rDHFR	0.74	0.73	9	0.09	75.49	2.56	0	0.70

Table 5.6 Metrics for QSAR networks with a minimum number of neighbours $k = 3$

difference in prediction error is not significant. In fact, in paired t-tests comparing the absolute prediction errors of Modular (OPLRA) $k = 0$ versus Modular (OPLRA) $k = 3$ for samples in all data set/data split tests, most p-values were above 0.30. The minimum p-value was $p = 0.08$ for the comparison in hDHFR data set and data split 4. In these tests, the applicability domain (AD) was defined in the same way as the original. The minimum number of neighbours is only enforced for training. Therefore, an unseen molecule without any neighbours in the trained graph is still considered to be outside the applicability domain.

Data Split	Data Set				
	NPYR1	NPYR2	CHRM3	hDHFR	rDHFR
Inside AD					
1	0.63 (±0.53)	0.55 (±0.54)	0.67 (±0.59)	0.71 (±0.62)	0.53 (±0.49)
2	0.61 (±0.63)	0.57 (±0.42)	0.68 (±0.74)	0.67 (±0.68)	0.64 (±0.63)
3	0.56 (±0.59)	0.47 (±0.41)	0.73 (±0.67)	0.62 (±0.67)	0.54 (±0.57)
4	0.44 (±0.44)	0.54 (±0.38)	0.56 (±0.55)	0.77 (±0.59)	0.59 (±0.59)
5	0.63 (±0.68)	0.59 (±0.61)	0.64 (±0.57)	0.68 (±0.66)	0.56 (±0.52)
Outside AD					
1	0.37 (±0.50)	0.69 (±0.99)	1.19 (±0.87)	0.58 (±0.00)	-
2	0.24 (±0.34)	0.43 (±0.29)	1.71 (±1.23)	2.67 (±0.68)	-
3	0.31 (±0.46)	0.36 (±0.32)	1.19 (±0.88)	-	-
4	0.50 (±1.16)	0.30 (±0.27)	2.26 (±1.91)	0.45 (±0.00)	0.64 (±0.57)
5	0.20 (±0.22)	0.34 (±0.53)	1.55 (±1.30)	1.43 (±1.94)	1.91 (±0.00)

Table 5.7 Performance of Modular (OPLRA) with $k = 3$ in the external set

Conclusions

Network representation is a useful tool for SAR analysis. Visualisation of molecular networks allows for a quick grasp of the homogeneity/heterogeneity of the data and module detection combined with the analysis of the common core of molecules helps to understand the context and characteristics of the available chemical space.

Modular (OPLRA) and OPLRAreg follow the same basic principle that QSAR predictions can be improved if modules are separated into informative clusters. But Modular (OPLRA) performs a more granular division of the data and this two-step clustering procedure was shown to be more predictive than OPLRAreg. Modular (OPLRA) modules were also more coherent and aligned with the structure similarity of molecules, which improves interpretation of the QSAR models.

Of course, one could have studied the PubChem assay text records directly and identified similar groupings for QSAR data but network analysis makes this task quicker and less laborious when first confronting a medium or large data set. The analysis described in this chapter showed that singletons will generally represent a diverse set of molecules and might indicate an initial effort made by researchers to map the chemical space in search of promising drugs or probes candidates. Dense modules, on the other hand, represent groups of molecules where usually a common core can be identified. An automatic workflow like Modular (OPLRA) can improve the discovery of these subgroups and help to visualise these complex relationships more easily.

Modular (OPLRA) facilitates the identification of groups of molecules with activity cliffs, with the potential to help medicinal chemists identify new promising paths for drug discovery more easily. The algorithm could also be used to generate

constraints for *de novo* drug design, contributing to lead optimisation of promising compounds.

Predictions in the external set were not affected when a minimum number of neighbours was enforced and network properties only changed in data sets that had a large number of singletons. The advantage of this approach lies on the reduced number of modules, which might facilitate the visualisation of network modules but at the cost of generating "artificial" links between non-similar compounds. This trade-off should be take into account when developing a new model and the requirement for the number of neighbours should probably be driven by the needs of specific QSAR projects. In future works, I would be interested in exploring the impact of other network construction techniques, with different fingerprints and similarity metrics, in the predictions of the method. Even in the current algorithm, it would be interesting to study the impact of the selected threshold in more details.

Other immediate extensions of the current work would be to study selectivity, instead of activity of compounds. It might be possible to identify modules with identifying fragments, substructures or descriptors that help to explain why certain compounds have more affinity for a specific receptor, say NPYR2, than others, e.g. NPYR1. Modular (OPLRA) could also be used to study selectivity of compounds for drug targets in specific organisms. One possible application would be to study the activity of inhibitors of DHFR in other organisms (e.g. *Candida albicanis*) compared to the human and rat models.

A few limitations of Modular (OPLRA) should also be addressed in future works. As it was shown in the network analysis section, most pairs of nodes are assigned to a common module even when samples building the network are resampled from the full data. But a small pair still does not get clustered together

consistently. It is important to understand the reasons for that, it might be that modularity is not the best metric for this application and it could be replaced by another metric or a technique of consensus clustering which produces clusters that generalise even better from a subsample to a full data set.

Chapter 6

Conclusions and future work

This thesis has investigated solutions to optimisation problems in several applications of complex data analysis, including the temporal evolution of groups in dynamic complex networks, piecewise linear models for QSAR and network techniques applied to QSAR. This final chapter discusses concluding remarks for the work presented in this thesis and point to directions of future works.

6.1 Concluding Remarks

Chapter 1 introduces the recent advantages in technology which has enabled the development of machine learning techniques and the need for more transparency in these models. This was followed by the research aims of this research project and the outline for this thesis.

Chapter 2 introduces the main interdisciplinary concepts related to the methods proposed in this thesis. The chapter introduced concepts of mathematical programming, network science and provided a brief introduction to quantitative

structure-activity relationships models and outlined the connection between these themes in the thesis.

In Chapter 3, SeqMod is introduced, a mathematical programming based algorithm for detecting the evolution of groups in complex networks. At each time step of a temporal network, SeqMod uses a previous reference snapshot of the temporal network to counterbalance the groups identified in the current snapshot, this technique effectively detected the ground truth of clusters in the datasets studied and performed better than similar algorithms.

Since the publication of the SeqMod paper (Chapter 3), there have been publications questioning the notion of ground truth. In this and in related work, it is assumed that node attributes correspond to the real groups and the goal of a community detection algorithm is to uncover that membership from the topology of networks. But this might not always be the case [188, 187]. The validation of network partitions will likely change in future years and, quite possibly, the definition of communities might need to accommodate these recent insights.

In Chapters 4 and 5, two different techniques were proposed to tackle QSAR models. The proposed algorithms are meant to be additional tools of QSAR analysis for medicinal chemists to model inhibitors of specific drug targets. These techniques could help these researchers explore new data sets and get a wider picture of their data sets. OPLRA and Modular (OPLRA) were validated using standard and robust validation procedures and were shown to produce predictions of similar accuracy of state of art algorithms and are an alternative to non-interpretable machine learning models commonly used in QSAR studies. Sensitive areas such as drug discovery, personalised medicine and health informatics could benefit more from techniques that better explain the relationship between inputs and output.

Modular (OPLRA), introduced in Chapter 5, is a first step towards using community detection in a network representation of these data sets to automatically create QSAR models. The fact that most nodes co-occur in the same module even when the data set is randomly sub-sampled indicates that the method is robust and modules in the sub-sampled network are still representative of the "real" modules detected when the full data set is available. But, of course, modules are not totally consistent and a few nodes that would normally be grouped in the full network can be assigned to different modules depending on the connectivity of the trained network. This is a limitation of community detection algorithms, in particular of modularity optimisation where the objective function exhibit degeneracy and an optimal modularity value could represent multiple valid solutions. But it is also an effect of the validation procedure.

6.2 Future work

This section makes suggestions on future research directions on the topics covered by this thesis.

With regards to community detection of temporal networks, a direct application of SeqMod is anomaly detection. The algorithm could be extended to identify cases when the modular structure deviates too much from previous time frames. The validation of algorithms and the notion of ground truth of network modules would also have to be investigated more deeply. This problem is not unique to dynamic networks, community detection applications for many network types have to be validated on networks with well defined and known clusters. One possible direction for research may be the generation of synthetic dynamic networks. If we succeed in generating synthetic networks with appropriate null models and

unambiguous clusters, validation of community detection algorithms will be done with greater ease and certainty.

Temporal networks are a specific type of multilayer networks and concepts of SeqMod and evolutionary clustering could be extended to handle the more generic cases. In dynamic multilayer networks, the extension is more straightforward but even in networks with multiple layers where no temporal dimension exist, a technique like SeqMod could be applied and provide insights about the complex nature of these relationships. Instead of using a past snapshot as temporal reference to cluster another network slice, we could, for example, investigate how the community structure in certain layers could or could not help to detect the communities in other layers.

With regards to QSAR applications, OPLRAreg and Modular (OPLRA) could be used for virtual screening. Large chemical databases like ChEMBL could be queried to identify compounds in the applicability domain of models built with these algorithms that have been tested against other enzymes or proteins and could possibly be repurposed. These techniques could also be tested using different classes of molecular descriptors to include 3D information and chirality of molecules. QSAR is essentially ligand-based but one could also extend these algorithms to include features related to the structure of the receptor as well, in the cases when that structure is known.

One related question that has not been directly addressed by this thesis and could be tackled in the future is: what is a module, after all? This fundamental question has many direct implications on the results of all algorithms presented here, including the QSAR models, and has been asked many times in the community detection literature. Even the most used and one of the most useful

mathematical representation, the modularity metric, cannot offer a single clear depiction of what the best modules in a network are.

The problem is that there are just too many possible ways to solve and think about clustering and communities. In fact, different experts could look at the same data or network and come to different conclusions about the perfect clustering for their data. New and inventive ways to answer this question would have to be sought in the future but, despite the current philosophical uncertainties and limitations, the methods and exploratory analysis described in this thesis are examples that data clustering can help us, even today, to predict and understand a bit better about phenomena in real complex systems of the world around us.

References

[1] Richard Cumbley and Peter Church. Is Big Data creepy? *Computer Law and Security Review*, 29:601–609, oct 2013. URL http://dx.doi.org/10.1016/j.clsr.2013.07.007.

[2] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, 47:98–115, 2015. ISSN 03064379. URL https://dx.doi.org/10.1016/j.is.2014.07.006.

[3] Cisco. *The Zettabyte Era: Trends and Analysis*. Cisco, 2017. URL https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html. Accessed: 2018-01-07.

[4] Alice E Williamson, Paul M Ylioja, Murray N Robertson, Vicky Avery, Jonathan B Baell, Harikrishna Batchu, Sanjay Batra, Jeremy N Burrows, Soumya Bhattacharyya, Felix Calderon, Susan A. Charman, Julie Clark, Benigno Crespo, Matin Dean, Stefan L Debbert, Michael Delves, Adelaide S.M. Dennis, Frederik Deroose, Sandra Duffy, Sabine Fletcher, Guri Giaever, Irene Hallyburton, Francisco Javier Gamo, Marinella Gebbia, R. Kiplin Guy, Zoe Hungerford, Kiaran Kirk, Maria J. Lafuente-Monasterio, Anna Lee, Stephan Meister, Corey Nislow, John P. Overington, George Papadatos, Luc Patiny, James Pham, Stuart A. Ralph, Andrea Ruecker, Eileen Ryan, Christopher Southan, Kumkum Srivastava, Chris Swain, Matthew J. Tarnowski, Patrick Thomson, Peter Turner, Iain M Wallace, Timothy N.C. Wells, Karen White, Laura White, Paul Willis, Elizabeth A Winzeler, Sergio Wittlin, Matthew H Todd, and Yevgeniya Antonova-Koch. Open source drug discovery: Highly potent antimalarial compounds derived from the tres cantos arylpyrroles. *ACS Central Science*, 2(10):687–701, 2016. URL http://dx.doi.org/10.1021/acscentsci.6b00086.

[5] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Stastical Learning: Data Mining, Inference, and Prediction*. Springer, 2 edition, 2009. URL http://dx.doi.org/10.1007/978-0-387-84858-7.

[6] Ian Witten, Eibe Frank, and Mark Hall. *Data Mining: Practical Machine Learning Tools and Techniques (Third Edition)*, volume 54. Morgan Kaufmann, 3 edition, 2011. URL https://www.sciencedirect.com/science/book/9780123748560. Accessed: 2018-01-07.

[7] Erik Brynjolfsson and Tom Mitchell. What can machine learning do? Workforce implications. *Science*, 358(6370):1530–1534, dec 2017. ISSN 0036-8075. URL http://dx.doi.org/10.1126/science.aap8062.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, pages 1–9, 2012. URL http://dx.doi.org/10.1016/j.protcy.2014.09.007.

[9] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, 29(6):82–97, nov 2012. URL http://dx.doi.org/10.1109/MSP.2012.2205597.

[10] Paul Sajda. Machine Learning for Detection and Diagnosis of Diseases. *Annual Review of Biomedical Engineering*, 8(1):537–565, aug 2006. URL http://dx.doi.org/10.1146/annurev.bioeng.8.061505.095802.

[11] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep Learning in Medical Image Analysis. *Annual Review of Biomedical Engineering*, 19(1):221–248, jun 2017. URL https://dx.doi.org/10.1146/annurev-bioeng-071516-044442.

[12] Alfredo Vellido, José D. Martín-Guerrero, and Paulo Lisboa. Making machine learning models interpretable. In *Proceedings of the 20th European Symposium on Artificial Neural Networks - ESANN 2012*, pages 163–172, Bruges, Belgium, 2012. URL https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2012-7.pdfs. Accessed: 2018-01-07.

[13] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?". In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pages 1135–1144, New York, New York, USA, feb 2016. ACM Press. URL http://dx.doi.org/10.1145/2939672.2939778.

[14] Adrien Bibal and Benoît Frénay. Interpretability of Machine Learning Models and Representations : an Introduction. In *Proceedings of the 24th European Symposium on Artificial Neural Networks - ESANN 2016*, pages 77–82, Bruges, Belgium, 2016. URL https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2016-141.pdf. Accessed: 2018-01-07.

[15] Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. "What is relevant in a text document?": An interpretable machine learning approach. *PLOS ONE*, 12(8):e0181142, aug 2017. URL http://dx.doi.org/10.1371/journal.pone.0181142.

[16] Grégoire Montavon, Wojciech Samek, and Klaus Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, feb 2018. URL http://dx.doi.org/10.1016/j.dsp.2017.10.011.

[17] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 2002. URL https://dx.doi.org/10.1103/RevModPhys.74.47.

[18] Steven H. Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, 2001. URL http://dx.doi.org/10.1038/35065725.

[19] Mark E. J. Newman. The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167–256, jan 2003. URL http://dx.doi.org/10.1137/S003614450342480.

[20] Sergey N. Dorogovtsev and José F. F. Mendes. Evolution of networks. *Advances in physics*, 2002. URL http://dx.doi.org/10.1080/00018730110112519.

[21] Gergely Palla, Albert-László Barabási, and Tamás Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–7, apr 2007. URL http://dx.doi.org/10.1038/nature05670.

[22] Petter Holme and Jari Saramäki. Temporal networks. *Physics Reports*, 519 (3):97–125, 2012. doi: 10.1016/j.physrep.2012.03.001. URL http://dx.doi.org/10.1016/j.physrep.2012.03.001.

[23] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, sep 2014. URL http://dx.doi.org/10.1093/comnet/cnu016.

[24] Stefano Boccaletti, G. Bianconi, R. Criado, C. I. del Genio, Jesús Gómez-Gardeñes, M. Romance, I. Sendiña-Nadal, Z. Wang, and M. Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1): 1–122, 2014. URL http://dx.doi.org/10.1016/j.physrep.2014.07.001.

[25] Laura Bennett, Aristotelis Kittas, Gareth Muirhead, Lazaros G. Papageorgiou, and Sophia Tsoka. Detection of composite communities in multiplex biological networks. *Scientific reports*, 5:10345, may 2015. URL http://dx.doi.org/10.1038/srep10345.

[26] Marc Barthélemy, Alain Barrat, Romualdo Pastor-Satorras, and Alessandro Vespignani. Characterization and modeling of weighted networks. In *Physica A: Statistical Mechanics and its Applications*, volume 346, pages 34–43, 2005. doi: 10.1016/j.physa.2004.08.047. URL http://dx.doi.org/10.1016/j.physa.2004.08.047.

[27] A Reka, Hawoong Jeong, Albert-László Barabási, Réka Albert, Hawoong Jeong, and Albert-László Barabási. L03_disc_Error and Attack Tolerance of Complex Networks. *Nature*, 406(July):378–381, 2000. doi: 10.1038/35019019. URL http://dx.doi.org/10.1038/35019019.

[28] Duncan J Watts. *Six degrees: The science of a connected age*. WW Norton & Company, 2004.

[29] Richard J Williams, Eric L Berlow, Jennifer A Dunne, Albert-László Barabási, and Neo D Martinez. Two degrees of separation in complex food webs. *Proceedings of the National Academy of Sciences of the United States of America*, 99(20):12913–6, oct 2002. URL http://dx.doi.org/10.1073/pnas.192448799.

[30] S. Havlin, D. Y. Kenett, E. Ben-Jacob, A. Bunde, R. Cohen, H. Hermann, J. W. Kantelhardt, J. Kertész, S. Kirkpatrick, J. Kurths, J. Portugali, and S. Solomon. Challenges in network science: Applications to infrastructures, climate, social systems and economics. *The European Physical Journal Special Topics*, 214(1):273–293, dec 2012. ISSN 1951-6355. URL http://dx.doi.org/10.1140/epjst/e2012-01695-x.

[31] Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, pages 425–443, 1969. URL http://dx.doi.org/10.2307/2786545.

[32] Ravi Kumar, Jasmine Novak, and A Tomkins. Structure and evolution of online social networks. *Link Mining: Models, Algorithms, and Applications*, pages 337–357, 2010. URL http://dx.doi.org/10.1145/1150402.1150476.

[33] Albert-László Barabási and R Albert. Emergence of scaling in random networks. *Science*, 286(5439):1–11, oct 1999. URL http://dx.doi.org/10.1126/science.286.5439.509.

[34] Albert-László Barabási. Scale-free networks: A decade and beyond. *Science*, 325(5939):412–413, 2009. URL http://dx.doi.org/10.1126/science.1173299.

[35] Daniel W Franks, Jason Noble, Peter Kaufmann, and Sigrid Stagl. Extremism propagation in social networks with hubs. *Adaptive Behavior*, 16(4):264–274, 2008. URL http://dx.doi.org/10.1177/1059712308090536.

[36] Albert-László Barabási, Natali Gulbahce, and Joseph Loscalzo. Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, 12(1):56–68, 2011. URL http://dx.doi.org/10.1038/nrg2918.

[37] Reuven Cohen, Keren Erez, Daniel Ben-Avraham, and Shlomo Havlin. Resilience of the internet to random breakdowns. *Physical review letters*, 85(21):4626, 2000. URL http://dx.doi.org/10.1103/PhysRevLett.85.4626.

[38] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):103, jun 2010. URL http://dx.doi.org/10.1016/j.physrep.2009.11.002.

[39] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1–44, nov 2016. URL http://dx.doi.org/10.1016/j.physrep.2016.09.002.

[40] B. W. Kernighan and S. Lin. An Efficient Heuristic Procedure for Partitioning Graphs. *Bell System Technical Journal*, 49:291–307, 1970. URL http://10.1002/j.1538-7305.1970.tb01770.x.

[41] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101:2658–2663, 2004. URL http://dx.doi.org/10.1073/pnas.0400054101.

[42] Brian Karrer and Mark E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, jan 2011. URL http://dx.doi.org/10.1103/PhysRevE.83.016107.

[43] Mark E. J. Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, pages 1–16, 2004. URL http://dx.doi.org/10.1103/PhysRevE.69.026113.

[44] Mark E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America (2006)*, 103(23):8577–82, jun 2006. URL http://dx.doi.org/10.1073/pnas.0601602103.

[45] Wayne W. Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977. URL http://www.jstor.org/stable/3629752. Accessed: 2018-01-05.

[46] Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random Structures & Algorithms*, 6(2-3):161–180, mar 1995. URL http://dx.doi.org/10.1002/rsa.3240060204.

[47] Michael Molloy and Bruce Reed. The Size of the Giant Component of a Random Graph with a Given Degree Sequence. *Combinatorics, Probability and Computing*, 7(3):295–305, sep 1998. URL http://dx.doi.org/10.1017/S0963548398003526.

[48] S. Boccaletti, V. Latora, Y Moreno, M. Chavez, and D. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4-5):175–308, February 2006. ISSN 03701573. URL http://dx.doi.org/10.1016/j.physrep.2005.10.009.

[49] Mark E. J. Newman. *Networks: an introduction*. Oxford university press, 2010. URL https://global.oup.com/academic/product/networks-9780199206650. Accessed: 2018-01-05.

[50] Santo Fortunato and Marc Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences of the United States of America*, 104(1):36–41, jan 2007. URL http://dx.doi.org/10.1073/pnas.0605965104.

[51] A. Arenas, A. Fernández, and S. Gómez. Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*, 10(5):053039, 2008. doi: 10.1088/1367-2630/10/5/053039.

[52] V. A. Traag, P. Van Dooren, and Y. Nesterov. Narrow scope for resolution-limit-free community detection. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 84:016114, Jul 2011. doi: 10.1103/PhysRevE.84.016114. URL http://dx.doi.org/10.1103/PhysRevE.84.016114.

[53] Boleslaw K. Szymanski Mingming Chen, Tommy Nguyen. A new metric quality of network community structure. *ASE Human Journal*, 2(4):226–240, 2013. URL https://www.cs.rpi.edu/~szymansk/papers/ase-human.13.pdf.

[54] Tianlong Chen, Pramesh Singh, and Kevin E Bassler. Network community detection using modularity density measures. *Journal of Statistical Mechanics: Theory and Experiment*, 2018(5):053406, 2018. URL https://doi.org/10.1088/1742-5468/aabfc8.

[55] Benjamin H. Good, Yves Alexandre De Montjoye, and Aaron Clauset. Performance of modularity maximization in practical contexts. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 81, 2010. URL http://dx.doi.org/10.1103/PhysRevE.81.046106.

[56] Ulrik Brandes, Daniel Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner. On Modularity Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188, feb 2008. URL http://dx.doi.org/10.1109/TKDE.2007.190689.

[57] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), October 2008. URL http://dx.doi.org/10.1088/1742-5468/2008/10/P10008.

[58] Mark E. J. Newman. Spectral methods for community detection and graph partitioning. *Physical Review E*, 88(4):042822, oct 2013. URL http://dx.doi.org/10.1103/PhysRevE.88.042822.

[59] Daniel Aloise, Gilles Caporossi, Pierre Hansen, Leo Liberti, Sylvain Perron, and Manuel Ruiz. Modularity maximization in networks by variable neighborhood search. In *Proceedings of the 10th DIMACS implementation challenge workshop*, pages 113–127, Atlanta, GA, USA, 2013. American Mathematical Society (AMS). URL http://dx.doi.org/10.1090/conm/588.

[60] Eslam Ali Hassan, Ahmed Ibrahem Hafez, Aboul Ella Hassanien, and Aly A Fahmy. Community detection algorithm based on artificial fish swarm optimization. In *Proceedings of the 7th IEEE International Conference Intelligent Systems IS'2014, Intelligent Systems'2014*, pages 509–521, Warsaw, Poland, 2015. Springer International Publishing. URL http://dx.doi.org/10.1007/978-3-319-11310-4_44.

[61] Roger Guimera and Luis A. Nunes Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900, 2005. URL http://dx.doi.org/10.1038/nature03288.

[62] Gang Xu, Sophia Tsoka, and Lazaros G. Papageorgiou. Finding community structures in complex networks using mixed integer optimisation. *The European Physical Journal B*, 60:231–239, 2007. URL http://dx.doi.org/10.1140/epjb/e2007-00331-0.

[63] G. Agarwal and D. Kempe. Modularity-maximizing graph communities via mathematical programming. *Eur. Phys. J. B*, 66:409–418, 2008. URL http://dx.doi.org/10.1140/epjb/e2008-00425-1.

[64] Gang Xu, Laura Bennett, Lazaros G Papageorgiou, and Sophia Tsoka. Module detection in complex networks using integer optimisation. *Algorithms for Molecular Biology*, 5(1):1–11, 2010. URL http://dx.doi.org/10.1186/1748-7188-5-36.

[65] Daniel Aloise, Sonia Cafieri, Gilles Caporossi, Pierre Hansen, Sylvain Perron, and Leo Liberti. Column generation algorithms for exact modularity maximization in networks. *Physical Review E*, 82:046112–046121, 2010. URL http://dx.doi.org/10.1103/PhysRevE.82.046112.

[66] Sonia Cafieri, Pierre Hansen, and Leo Liberti. Locally optimal heuristic for modularity maximization of networks. *Physical Review E*, 83:056105, 2011. URL http://dx.doi.org/10.1103/PhysRevE.83.056105.

[67] Roger Fletcher. *Practical Methods of Optimization*. Wiley, New Jersey, USA, 2 edition, 2013. URL https://www.wiley.com/en-sg/Practical+Methods+of+Optimization,+2nd+Edition-p-9780471494638. Accessed: 2018-01-07.

[68] Junyi Chai, James N.K. Liu, and Eric W.T. Ngai. Application of decision-making techniques in supplier selection: A systematic review of literature. *Expert Systems with Applications*, 40(10):3872–3885, 2013. URL http://dx.doi.org/10.1016/j.eswa.2012.12.040.

[69] Josefa Mula, David Peidro, Manuel Díaz-Madroñero, and Eduardo Vicens. Mathematical programming models for supply chain production and transport planning. *European Journal of Operational Research*, 204(3):377–390, 2010. URL http://dx.doi.org/10.1016/j.ejor.2009.09.008.

[70] X. Xia and A.M. Elaiw. Optimal dynamic economic dispatch of generation: A review. *Electric Power Systems Research*, 80(8):975–986, 2010. URL http://dx.doi.org/10.1016/j.epsr.2009.12.012.

[71] Chrysanthi Ainali, Frank Nestle, Lazaros G. Papageorgiou, and Sophia Tsoka. Disease classification through integer optimisation. In *21st European Symposium on Computer Aided Process Engineering*, volume 29 of *Computer Aided Chemical Engineering*, pages 1548 – 1552. Elsevier, 2011. URL https://dx.doi.org/10.1016/B978-0-444-54298-4.50088-X.

[72] S.X. Chen, H.B. Gooi, and M.Q. Wang. Sizing of energy storage for microgrids. *IEEE Transactions on Smart Grid*, 3(1):142–151, 2012. URL http://dx.doi.org/10.1109/TSG.2011.2160745.

[73] Y. Xiao, Q. Zhao, I. Kaku, and Y. Xu. Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers and Operations Research*, 39(7):1419–1431, 2012. URL http://dx.doi.org/10.1016/j.cor.2011.08.013.

[74] Sonia Cafieri and David Rey. Maximizing the number of conflict-free aircraft using mixed-integer nonlinear programming. *Computers and Operations Research*, 80:147 – 158, 2017. ISSN 0305-0548. URL https://dx.doi.org/10.1016/j.cor.2016.12.002.

[75] Liu Xiang, Jun Luo, and Athanasios Vasilakos. Compressed data aggregation for energy efficient wireless sensor networks. In *Proceedings of the 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks - SECON 2011*, pages 46–54, Salt Lake City, UT, USA, 2011. URL http://dx.doi.org/10.1109/SAHCN.2011.5984932.

[76] Richard E. Rosenthal. *GAMS — A User's Guide*. GAMS Development Corporation, Washington, DC, USA, 2017. URL https://www.gams.com/24.8/docs/userguides/GAMSUsersGuide.pdf. Accessed: 2018-01-05.

[77] Robert Fourer, David M Gay, and Brian Kernighan. *Ampl*. Cengage Learning, 2 edition, 2002. URL https://ampl.com/resources/the-ampl-book. Accessed: 2018-01-05.

[78] Johannes Bisschop. *AIMMS optimization modeling*. AIMMS B.V., 2017. URL https://aimms.com/english/developers/resources/manuals/optimization-modeling/. Accessed: 2018-01-05.

[79] Stefan Theussl and Hans W. Borchers. The r project for statistical computing | cran task view: Optimization and mathematical programming, Nov 2017. URL https://cran.r-project.org/web/views/Optimization.html. Accessed: 2018-01-05.

[80] William E. Hart, Jean-Paul Watson, and David L. Woodruff. Pyomo: modeling and solving mathematical programs in python. *Mathematical Programming Computation*, 3(3):219–260, 2011. URL https://doi.org/10.1007/s12532-011-0026-8.

[81] International Business Machines Corporation. Ibm ilog cplex v12. 7: User's manual for cplex, 2009. URL https://www.ibm.com/support/knowledgecenter/SSSA5P_12.7.0/ilog.odms.studio.help/pdf/usrcplex.pdf.

[82] Christian Bliek1ú, Pierre Bonami, and Andrea Lodi. Solving mixed-integer quadratic programming problems with ibm-cplex: a progress report. In *Proceedings of the 26th RAMP Symposium*, pages 16–17, Tokyo, Japan, 2014. URL http://www.orsj.or.jp/ramp/2014/paper/4-3.pdf. Accessed: 2018-01-05.

[83] Gurobi Optimization. Gurobi optimizer reference manual, 2015. URL http://www.gurobi.com. Accessed: 2018-01-05.

[84] Robin Lougee-Heimer. The common optimization interface for operations research: Promoting open-source software in the operations research community. *IBM Journal of Research and Development*, 47(1):57–66, 2003. URL http://dx.doi.org/10.1147/rd.471.0057.

[85] Laura Bennett, Songsong Liu, Lazaros G. Papageorgiou, and Sophia Tsoka. Detection of Disjoint and Overlapping Modules in Weighted Complex Networks. *Advances in Complex Systems*, 15(05):1150023, jul 2012. URL http://dx.doi.org/10.1142/S0219525911500238.

[86] Lingjian Yang, Jonathan C. Silva Silva, Lazaros G. Papageorgiou, and Sophia Tsoka. Community Structure Detection for Directed Networks through Modularity Optimisation. *Algorithms*, 9(4):73, 2016. URL http://dx.doi.org/10.3390/A9040073.

[87] Laura Bennett. *Community Structure Detection in Complex Biological Networks*. Phd thesis, King's College London, 2012. URL https://kclpure.kcl.ac.uk/portal/files/12490434/Studentthesis-Laura_Bennett_2013.pdf. Accessed: 2018-01-05.

[88] Laura Bennett, Aristotelis Kittas, Songsong Liu, Lazaros G. Papageorgiou, and Sophia Tsoka. Community structure detection for overlapping modules through mathematical programming in protein interaction networks. *PLoS ONE*, 9(11), 11 2014. doi: 10.1371/journal.pone.0112821. URL http://dx.doi.org/10.1371%2Fjournal.pone.0112821.

[89] Sonia Cafieri, Alberto Costa, and Pierre Hansen. Reformulation of a model for hierarchical divisive graph modularity maximization. *Annals of Operations Research*, 222(1):213–226, 2014. URL https://dx.doi.org/10.1007/s10479-012-1286-z.

[90] Alberto Costa. Milp formulations for the modularity density maximization problem. *European Journal of Operational Research*, 245(1):14–21, 2015. URL https://dx.doi.org/10.1016/j.ejor.2015.03.012.

[91] Rafael Santiago and Luís C. Lamb. Efficient modularity density heuristics for large graphs. *European Journal of Operational Research*, 258(3):844 – 865, 2017. URL https://dx.doi.org/10.1016/j.ejor.2016.10.033.

[92] Rafael de Santiago and Luís C. Lamb. Exact computational solution of modularity density maximization by effective column generation. *Computers and Operations Research*, 86:18 – 29, 2017. URL https://dx.doi.org/10.1016/j.cor.2017.04.013.

[93] Jure Leskovec, Kevin J. Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web - WWW '10*, page 631, New York, New York, USA, 2010. ACM Press. URL http://dx.doi.org/10.1145/1772690.1772755.

[94] Chuan Shi, PS Yu, Y Cai, Z Yan, and Bin Wu. On selection of objective functions in multi-objective community detection. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, volume 1, pages 2301–2304, 2011. URL https://dx.doi.org/10.1145/2063576.2063951.

[95] Yen-Chuen Wei and Chung-Kuan Cheng. Towards efficient hierarchical designs by ratio cut partitioning. In *1989 IEEE International Conference on Computer-Aided Design. Digest of Technical Papers*, pages 298–301, Santa Clara, CA, USA, 1989. IEEE Comput. Soc. Press. URL http://dx.doi.org/10.1109/ICCAD.1989.76957.

[96] Maoguo Gong, Xiaowei Chen, Lijia Ma, Qingfu Zhang, and Licheng Jiao. Identification of multi-resolution network structures with multi-objective immune algorithm. *Applied Soft Computing*, 13(4):1705–1717, April 2013. URL https://dx.doi.org/10.1016/j.asoc.2013.01.018.

[97] L Angelini, S Boccaletti, D Marinazzo, M Pellicoro, and S Stramaglia. Identification of network modules by optimization of ratio association. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 17(2):023114, June 2007. URL https://dx.doi.org/10.1063/1.2732162.

[98] Clara Pizzuti. GA-Net: A genetic algorithm for community detection in social networks. In *Parallel Problem Solving from Nature PPSN X*, volume 5199, pages 1081–1090, Italy, 2008. Springer Berlin Heidelberg. URL http://dx.doi.org/10.1007/978-3-540-87700-4_107.

[99] Zhenping Li, Shihua Zhang, Rui Sheng Wang, Xiang Sun Zhang, and Luonan Chen. Quantitative function for community detection. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 77, 2008. doi: http://dx.doi.org/10.1103/PhysRevE.77.036109.

[100] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, pages 1–6, 2008. URL https://dx.doi.org/10.1103/PhysRevE.78.046110.

[101] Nathan Brown. Chemoinformatics—an introduction for computer scientists. *ACM Computing Surveys*, 41(2):1–38, 2009. URL http://dx.doi.org/10.1145/1459352.1459353.

[102] Jean-Louis Reymond, Lars Ruddigkeit, Lorenz Blum, and Ruud van Deursen. The enumeration of chemical space. *Wiley Interdisciplinary Reviews-Computational Molecular Science*, 2(5):717–733, 2012. URL http://dx.doi.org/10.1002/wcms.1104.

[103] William L Jorgensen. The many roles of computation in drug discovery. *Science*, 303(5665):1813–1818, 2004. ISSN 0036-8075. URL http://dx.doi.org/10.1126/science.1096361.

[104] Andrea R Beccari, Carlo Cavazzoni, Claudia Beato, and Gabriele Costantino. LiGen: a High Performance workflow for chemistry driven de novo design. *Journal of chemical information and modeling*, 53(6):1518–1527, 2013. URL http://dx.doi.org/10.1021/ci400078g.

[105] Samuel S Y Wong, Weimin Luo, and Keith C C Chan. EvoMD: an algorithm for evolutionary molecular design. *IEEE/ACM transactions on computational biology and bioinformatics*, 8(4):987–1003, 2011. URL http://dx.doi.org/10.1109/TCBB.2010.100.

[106] Nathan Brown, Ben McKay, François Gilardoni, and Johann Gasteiger. A Graph-Based Genetic Algorithm and Its Application to the Multiobjective Evolution of Median Molecules. *Journal of Chemical Information and Computer Sciences*, 44(3):1079–1087, may 2004. URL http://dx.doi.org/10.1021/ci034290p.

[107] Christos A. Nicolaou and Nathan Brown. Multi-objective optimization methods in drug design. *Drug Discovery Today: Technologies*, 10(3):e427–e435, 2013. URL http://dx.doi.org/10.1016/j.ddtec.2013.02.001.

[108] Alexander Tropsha. Best Practices for QSAR Model Development, Validation, and Exploitation. *Molecular Informatics*, 29(6-7):476–488, 2010. URL http://dx.doi.org/10.1002/minf.201000061.

[109] Cleber C. Melo-Filho, Rafael F. Dantas, Rodolpho C. Braga, Bruno J. Neves, Mario R. Senger, Walter C G Valente, Joao M. Rezende-Neto, Willian T. Chaves, Eugene N. Muratov, Ross A. Paveley, Nicholas Furnham, Lee Kamentsky, Anne E. Carpenter, Floriano P. Silva-Junior, and Carolina H. Andrade. QSAR-Driven Discovery of Novel Chemical Scaffolds Active against Schistosoma mansoni. *Journal of Chemical Information and Modeling*, 56 (7):1357–1372, 2016. URL http://dx.doi.org/10.1021/acs.jcim.6b00055.

[110] Han Van De Waterbeemd and Eric Gifford. Admet in silico modelling: towards prediction paradise? *Nature reviews Drug discovery*, 2(3), 2003. URL http://dx.doi.org/10.1038/nrd1032.

[111] Richard A. Lewis. A general method for exploiting qsar models in lead optimization. *Journal of Medicinal Chemistry*, 48(5):1638–1648, 2005. URL http://dx.doi.org/10.1021/jm049228d.

[112] Nathan Brown and Richard A Lewis. Exploiting qsar methods in lead optimization. *Current opinion in drug discovery & development*, 9(4):419–424, 2006.

[113] Gautier Moroy, Virginie Y Martiny, Philippe Vayer, Bruno O Villoutreix, and Maria A Miteva. Toward in silico structure-based admet prediction in drug discovery. *Drug discovery today*, 17(1):44–55, 2012. URL http://dx.doi.org/10.1016/j.drudis.2011.10.023.

[114] Kathia M Honorio, Tiago L Moda, and Adriano D Andricopulo. Pharmacokinetic properties and in silico adme modeling in drug discovery. *Medicinal Chemistry*, 9(2):163–176, 2013. URL http://dx.doi.org/10.2174/1573406411309020002.

[115] Marcelo N. Gomes, Rodolpho C. Braga, Edyta M. Grzelak, Bruno J. Neves, Eugene N. Muratov, Rui Ma, Larry K. Klein, Sanghyun Cho, Guilherme R. Oliveira, Scott G. Franzblau, and Carolina Horta Andrade. QSAR-driven Design, Synthesis and Discovery of Potent and Selective Chalcone Derivatives with Antitubercular Activity. *European Journal of Medicinal Chemistry*, 137: 126–138, sep 2017. URL http://dx.doi.org/10.1016/j.ejmech.2017.05.026.

[116] Antonio Rescifina, Giuseppe Floresta, Agostino Marrazzo, Carmela Parenti, Orazio Prezzavento, Giovanni Nastasi, Maria Dichiara, and Emanuele Amata. Development of a Sigma-2 Receptor affinity filter through a Monte Carlo based QSAR analysis. *European Journal of Pharmaceutical Sciences*, 106: 94–101, aug 2017. URL http://dx.doi.org/10.1016/j.ejps.2017.05.061.

[117] Corwin Hansch, Peyton P. Maloney, Toshio Fujita, and Robert M. Muir. Correlation of biological activity of phenoxyacetic acids with Hammett

substituent constants and partition coefficients. *Nature*, 194(4824):178–180, 1962. URL http://dx.doi.org/10.1038/194178b0.

[118] Corwin Hansch, Robert M. Muir, Toshio Fujita, Peyton P. Maloney, Fred Geiger, and Margaret Streich. The Correlation of Biological Activity of Plant Growth Regulators and Chloromycetin Derivatives with Hammett Constants and Partition Coefficients. *Journal of the American Chemical Society*, 85 (18):2817–2824, 1963. URL http://dx.doi.org/10.1021/ja00901a033.

[119] Corwin Hansch, John E. Quinlan, and Gary L. Lawrence. Linear free-energy relationship between partition coefficients and the aqueous solubility of organic liquids. *The Journal of Organic Chemistry*, 33(1):347–350, 1968. URL http://dx.doi.org/10.1021/jo01265a071.

[120] Corwin Hansch and Carlo Silipoj. Quantitative Structure-Activity Relationship of Reversible Dihydrofolate Reductase Inhibitors. *Journal of Medicinal Chemistry*, 17(7):661–667, 1974. URL http://dx.doi.org/10.1007/BF00125376.

[121] T. A. Andrea and Hooshmand Kalayeh. Applications of neural networks in quantitative structure-activity relationships of dihydrofolate reductase inhibitors. *Journal of medicinal chemistry*, 34(9):2824–36, 1991. URL http://dx.doi.org/10.1021/jm00113a022.

[122] John B. O. Mitchell. Machine learning methods in chemoinformatics. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 4(October), sep 2014. URL http://dx.doi.org/10.1002/wcms.1183.

[123] Joseph Leonard and Kunal Roy. Comparative classical QSAR modeling of Ant-HIV Thiocarbamates. *QSAR & Combinatorial Science*, 26(9):980–990, 2007. URL http://dx.doi.org/10.1002/qsar.200630140.

[124] Yang Zhou, Zhong Ni, Keping Chen, Haijun Liu, Liang Chen, Chaoqun Lian, and Lirong Yan. Modeling protein-peptide recognition based on classical quantitative structure-affinity relationship approach: Implication for proteome-wide inference of peptide-mediated interactions. *Protein Journal*, 32(7):568–578, oct 2013. URL http://dx.doi.org/10.1007/s10930-013-9519-9.

[125] Maryam Salahinejad, Tu C. Le, and David A Winkler. Aqueous solubility prediction: Do crystal lattice interactions help? *Molecular Pharmaceutics*, 10(7):2757–2766, 2013. URL http://dx.doi.org/10.1021/mp4001958.

[126] Freya Klepsch, Poongavanam Vasanthanathan, and Gerhard F Ecker. Ligand and Structure-Based Classification Models for Prediction of P-Glycoprotein Inhibitors. *Journal of Chemical Information and Modeling*, 54(1):218–229, jan 2014. URL http://dx.doi.org/10.1021/ci400289j.

[127] Sedat Karabulut, Natalia Sizochenko, Adnan Orhan, and Jerzy Leszczynski. A DFT-based QSAR study on inhibition of human dihydrofolate reductase. *Journal of Molecular Graphics and Modelling*, 70:23–29, nov 2016. URL http://10.1016/j.jmgm.2016.09.005.

[128] Igor V Tetko, Daniel M Lowe, and Antony J Williams. The development of models to predict melting and pyrolysis point data associated with several hundred thousand compounds mined from PATENTS. *Journal of cheminformatics*, 8(1):2, 2016. URL http://dx.doi.org/10.1186/s13321-016-0113-y.

[129] Isidro Cortes-Ciriano and Andreas Bender. Improved Chemical Structure-Activity Modeling Through Data Augmentation. *Journal of Chemical Information and Modeling*, 55(12):2682–2692, 2015. URL http://dx.doi.org/10.1021/acs.jcim.5b00570.

[130] LiMin Fu. Rule learning by searching on adapted nets. In *Proceedings of the Ninth National Conference on Artificial Intelligence - AAAI-91*, pages 590 – 595, Anaheim, California, USA, 1991. URL https://www.aaai.org/Papers/AAAI/1991/AAAI91-092.pdf. Accessed: 2018-01-05.

[131] Johan Huysmans, Bart Baesens, and Jan Vanthienen. Using Rule Extraction to Improve the Comprehensibility of Predictive Models. *SSRN Electronic Journal*, 2006. URL http://dx.doi.org/10.2139/ssrn.961358.

[132] Pavel Polishchuk. Interpretation of Quantitative Structure-Activity Relationship Models: Past, Present, and Future. *Journal of Chemical Information and Modeling*, 57(11):2618–2639, 2017. URL http://dx.doi.org/10.1021/acs.jcim.7b00274.

[133] Toshio Fujita and David A. Winkler. Understanding the Roles of the "Two QSARs". *Journal of Chemical Information and Modeling*, page 269–274, 2016. URL http://dx.doi.org/10.1021/acs.jcim.5b00229.

[134] Victor E. Kuz'min, Pavel G. Polishchuk, Anatoly G. Artemenko, and Sergey A. Andronati. Interpretation of QSAR models based on random forest methods. *Molecular Informatics*, 30(6-7):593–603, 2011. URL http://dx.doi.org/10.1002/minf.201000173.

[135] Pavel G. Polishchuk, Victor E. Kuźmin, Anatoly G. Artemenko, and Eugene N. Muratov. Universal approach for structural interpretation of qsar/ qspr models. *Molecular Informatics*, 32(9-10):843–853, 2013. URL http://dx.doi.org/10.1002/minf.201300029.

[136] Vinicius M. Alves, Eugene N. Muratov, Stephen J. Capuzzi, Regina Politi, Yen Low, Rodolpho C. Braga, Alexey V. Zakharov, Alexander Sedykh, Elena Mokshyna, Sherif Farag, Carolina H. Andrade, Victor E. Kuz'min, Denis Fourches, and Alexander Tropsha. Alarms about structural alerts. *Green Chem.*, 18(16):4348–4360, 2016. URL http://dx.doi.org/10.1039/C6GC01492E.

[137] Chun Wei Yap. PaDEL-descriptor: An open source software to calculate molecular descriptors and fingerprints. *Journal of Computational Chemistry*, 32(7):1466–1474, may 2011. URL http://dx.doi.org/10.1002/jcc.21707.

[138] Christoph Steinbeck, Yongquan Han, Stefan Kuhn, Oliver Horlacher, Edgar Luttmann, and Egon Willighagen. The Chemistry Development Kit (CDK): An open-source Java library for chemo- and bioinformatics. *Journal of Chemical Information and Computer Sciences*, 43(2):493–500, 2003. URL http://dx.doi.org/10.1021/ci025584y.

[139] RDKit. *RDKit: Open-source cheminformatics*, 2017. URL http://www.rdkit.org. Accessed: 2018-01-05.

[140] Andrea Mauri, Viviana Consonni, Manuela Pavan, and Roberto Todeschini. Dragon software: An easy approach to molecular descriptor calculations. *MATCH Communications in Mathematical and in Computer Chemistry*, 56(2):237–248, 2006. URL http://match.pmf.kg.ac.rs/electronic_versions/Match56/n2/match56n2_237-248.pdf. Accessed: 2018-01-05.

[141] Santiago Vilar, Giorgio Cozza, and Stefano Moro. Medicinal chemistry and the molecular operating environment (moe): application of qsar and molecular docking to drug discovery. *Current topics in medicinal chemistry*, 8 (18):1555–1572, 2008. URL http://dx.doi.org/10.2174/156802608786786624.

[142] Kunal Roy, Supratik Kar, and Rudra Narayan Das, editors. *Understanding the Basics of QSAR for Applications in Pharmaceutical Sciences and Risk Assessment*. Academic Press, Boston, USA, 2015. URL https://doi.org/10.1016/B978-0-12-801505-6.00002-8.

[143] Lowell H. Hall and Lemont B. Kier. Electrotopological State Indices for Atom Types: A Novel Combination of Electronic, Topological, and Valence State Information. *Journal of Chemical Information and Modeling*, 35(6): 1039–1045, nov 1995. URL http://dx.doi.org/10.1021/ci00028a014.

[144] Darko Butina. Performance of Kier-Hall E-state descriptors in quantitative structure activity relationship (QSAR) studies of multifunctional molecules. *Molecules*, 9(12):1004–1009, 2004. URL http://dx.doi.org/10.3390/91201004.

[145] David J. Livingstone. The Characterization of Chemical Structures Using Molecular Properties. A Survey. *Journal of Chemical Information and Computer Sciences*, 40(2):195–209, mar 2000. URL http://dx.doi.org/10.1021/ci990162i.

[146] Isabelle Guyon and Andre Elisseeff. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003. URL http://www.jmlr.org/papers/volume3/guyon03a/guyon03a.pdf. Accessed: 2018-01-05.

[147] Joanna S. Jaworska, M. Comber, C. Auer, and CJ. Van Leeuwen. Summary of a workshop on regulatory acceptance of (q) sars for human health and environmental endpoints. *Environmental Health Perspectives*, 111(10):1358, 2003. URL http://www.jstor.org/stable/3435408.

[148] J. C. Dearden, Mark T. D. Cronin, and Klaus L. E. Kaiser. How not to develop a quantitative structure-activity or structure-property relationship (QSAR/QSPR). *SAR and QSAR in environmental research*, 20(February): 241–266, 2009. URL http://dx.doi.org/10.1080/10629360902949567.

[149] Dagmar Stumpfe and Jürgen Bajorath. Similarity searching. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(2):260–282, mar 2011. URL http://dx.doi.org/10.1002/wcms.23.

[150] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010. URL http://dx.doi.org/10.1021/ci100050t.

[151] Dávid Bajusz, Anita Rácz, and Károly Héberger. Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics*, 7(1), 2015. URL http://dx.doi.org/10.1186/s13321-015-0069-3.

[152] Jürgen Bajorath. Representation and identification of activity cliffs. *Expert Opinion on Drug Discovery*, pages 1–5, jul 2017. URL http://dx.doi.org/10.1080/17460441.2017.1353494.

[153] Gerald M. Maggiora. On outliers and activity cliffs - Why QSAR often disappoints. *Journal of Chemical Information and Modeling*, 46(4):1535, 2006. URL http://dx.doi.org/10.1021/ci060117s.

[154] Dagmar Stumpfe and Jürgen Bajorath. Exploring Activity Cliffs in Medicinal Chemistry. *Journal of Medicinal Chemistry*, 55(7):2932–2942, apr 2012. URL http://dx.doi.org/10.1021/jm201706b.

[155] Dagmar Stumpfe, Ye Hu, Dilyana Dimova, and Jürgen Bajorath. Recent Progress in Understanding Activity Cliffs and Their Utility in Medicinal Chemistry. *Journal of Medicinal Chemistry*, 57(1):18–28, jan 2014. URL http://dx.doi.org/10.1021/jm401120g.

[156] Dagmar Stumpfe, Dilyana Dimova, and Jürgen Bajorath. Composition and topology of activity cliff clusters formed by bioactive compounds. *Journal of Chemical Information and Modeling*, 54(2):451–461, 2014. URL http://dx.doi.org/10.1021/ci400728r.

[157] Mathias Wawer, Lisa Peltason, Nils Weskamp, Andreas Teckentrup, and Jurgen Bajorath. Structure–Activity Relationship Anatomy by Network-like Similarity Graphs and Local Structure–Activity Relationship Indices. *Journal of Medicinal Chemistry*, 51(19):6075–6084, oct 2008. URL http://dx.doi.org/10.1021/jm800867g.

[158] Vigneshwaran Namasivayam, Disha Gupta-Ostermann, Jenny Balfer, Kathrin Heikamp, and Jürgen Bajorath. Prediction of Compounds in Different Local Structure–Activity Relationship Environments Using Emerging Chemical Patterns. *Journal of Chemical Information and Modeling*, 54 (5):1301–1310, may 2014. URL http://dx.doi.org/10.1021/ci500147b.

[159] Michelle Girvan and Mark E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–6, June 2002. URL http://dx.doi.org/10.1073/pnas.122653799.

[160] Teresa M. Przytycka, Mona Singh, and Donna K. Slonim. Toward the dynamic interactome: It's about time. *Briefings in Bioinformatics*, 11(1): 15–29, 2010. URL dx.doi.org/10.1093/bib/bbp057.

[161] Thomas Aynaud and Jean-Loup Guillaume. Multi-step community detection and hierarchical time segmentation in evolving networks. In *Proceedings of the Fifth SNA-KDD Workshop Social Network Mining and Analysis, in conjunction with the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2011)*, volume 11, 2011.

[162] Andrea Lancichinetti and Santo Fortunato. Consensus clustering in complex networks. *Scientific Reports*, 2(336), March 2012. URL http://dx.doi.org/10.1038/srep00336.

[163] Chayant Tantipathananandh, Tanya Berger-Wolf, and David Kempe. A framework for community identification in dynamic social networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*, pages 717–726, New York, New York, USA, 2007. ACM Press. URL http://dx.doi.org/10.1145/1281192.1281269.

[164] Daniel J. Fenn, Mason a. Porter, Mark McDonald, Stacy Williams, Neil F. Johnson, and Nick S. Jones. Dynamic communities in multichannel data: An application to the foreign exchange market during the 2007-2008 credit crisis. *Chaos*, 19(3):1–8, 2009. URL http://dx.doi.org/10.1063/1.3184538.

[165] Julie Kauffman, Aristotelis Kittas, Laura Bennett, and Sophia Tsoka. Dyconet: A gephi plugin for community detection in dynamic complex networks. *PLoS ONE*, 9(7), July 2014. URL http://dx.doi.org/10.1371/journal.pone.0101357.

[166] Jimeng Sun, Philip S Yu, and Christos Faloutsos. GraphScope : Parameter-free Mining of Large Time-evolving Graphs. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*, pages 687–696, 2007. URL http://dx.doi.org/10.1145/1281192.1281266.

[167] Sitaram Asur, Srinivasan Parthasarathy, and Duygu Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs. *ACM Transactions on Knowledge Discovery from Data*, 3(4):1–36, November 2009. URL http://dx.doi.org/10.1145/1631162.1631164.

[168] Deepayan Chakrabarti, Ravi Kumar, and Andrew Tomkins. Evolutionary clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*, pages 554–560, New York, New York, USA, 2006. ACM Press. URL http://dx.doi.org/10.1145/1150402.1150467.

[169] Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*, pages 153–162, New York, New York, USA, 2007. ACM Press. URL http://dx.doi.org/10.1145/1281192.1281212.

[170] Min-Soo Kim and Jiawei Han. A particle-and-density based evolutionary clustering method for dynamic networks. *Proceedings of the VLDB Endowment*, 2(1):622–633, August 2009. URL http://dx.doi.org/10.14778/1687627.1687698.

[171] Yu-Ru Lin, Yun Chi, Shenghuo Zhu, Hari Sundaram, and Belle L. Tseng. Analyzing communities and their evolutions in dynamic social networks. *ACM Transactions on Knowledge Discovery from Data*, 3(2):1–31, 2009. URL http://dx.doi.org/10.1145/1514888.1514891.

[172] Nam P. Nguyen, Thang N. Dinh, Ying Xuan, and My T. Thai. Adaptive algorithms for detecting community structure in dynamic social networks. *Proceedings IEEE INFOCOM 2011*, pages 2282–2290, 2011. URL http://dx.doi.org/10.1109/INFCOM.2011.5935045.

[173] Robert Görke, Pascal Maillard, Andrea Schumm, Christian Staudt, and Dorothea Wagner. Dynamic graph clustering combining modularity and smoothness. *J. Exp. Algorithmics*, 18(1), 2011. URL http://dx.doi.org/10.1145/2444016.2444021.

[174] Vikas Kawadia and Sameet Sreenivasan. Sequential detection of temporal communities by estrangement confinement. *Scientific reports*, 2(794), 2012. URL http://dx.doi.org/10.1038/srep00794.

[175] Francesco Folino and Clara Pizzuti. An Evolutionary Multiobjective Approach for Community Discovery in Dynamic Networks. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1838–1852, August 2014. URL http://dx.doi.org/10.1109/TKDE.2013.131.

[176] Nam P Nguyen, Thang N Dinh, Yilin Shen, and My T Thai. Dynamic social community detection and its applications. *PloS one*, 9(4), January 2014. URL http://dx.doi.org/10.1371/journal.pone.0091431.

[177] Danielle S. Bassett, Mason A. Porter, Nicholas F. Wymbs, Scott T. Grafton, Jean M. Carlson, and Peter J. Mucha. Robust detection of dynamic community structure in networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 23(1):013142, 2013. URL http://dx.doi.org/10.1063/1.4790830.

[178] Julie Fournet and Alain Barrat. Contact patterns among high school students. *PloS one*, 9(9), January 2014. URL http://dx.doi.org/10.1371/journal.pone.0107878.

[179] Wen Dong, Bruno Lepri, and A. Pentland. Modeling the co-evolution of behaviors and social relationships using mobile phone data. In *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia - MUM '11*, pages 134–143, New York, New York, USA, 2011. ACM Press. URL http://dx.doi.org/10.1145/2107596.2107613.

[180] Vitor Baptista, Fernando Brito, Jansepetrus Brasileiro, Alexandre Nobrega Duarte, Ed Porto Bezerra, Filipe Almeida, Patricia Lima, and Samara Guimaraes. Uma ferramenta para analisar mudanças na coesão entre parlamentares em votações nominais. In *III Brazilian Workshop on Social Network Analysis and Mining*, pages 1–7, 2014. URL http://dx.doi.org/10.13140/2.1.2467.5201. (In Portuguese).

[181] Peter J. Mucha, Thomas Richardson, Kevin Macon, Mason A. Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, May 2010. URL http://dx.doi.org/10.1126/science.1184819.

[182] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, December 1985. URL http://dx.doi.org/10.1007/BF01908075.

[183] Marina Meilă. Comparing clusterings-an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007. URL http://dx.doi.org/10.1016/j.jmva.2006.11.013.

[184] Chris Fraley, Adrian E. Raftery, Thomas Brendan Murphy, and Luca Scrucca. *mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation*, 2012.

[185] Kevin S. Xu, Mark Kliger, and Alfred O. Hero. Evolutionary spectral clustering with adaptive forgetting factor. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2174–2177. IEEE, 2010. URL http://dx.doi.org/10.1109/ICASSP.2010.5495655.

[186] The Guardian. Petrobras scandal: Brazilian oil executives among 35 charged. http://goo.gl/fqUaqY, 2015. Accessed: 2015-10-06.

[187] Mark E. J. Newman and Aaron Clauset. Structure and inference in annotated networks. *Nature Communications*, 7:11863, jun 2016. URL http://dx.doi.org/10.1038/ncomms11863.

[188] Vasyl Palchykov, Valerio Gemmetto, Alexey Boyarsky, and Diego Garlaschelli. Ground truth? Concept-based communities versus the external classification of physics manuscripts. *EPJ Data Science*, 5(1):28, dec 2016. URL http://dx.doi.org/10.1140/epjds/s13688-016-0090-4.

[189] Darko Hric, Tiago P Peixoto, and Santo Fortunato. Network structure, metadata, and the prediction of missing nodes and annotations. *Physical Review X*, 6(3):031038, 2016. URL https://doi.org/10.1103/PhysRevX.6.031038.

[190] Xiao Zhang, Cristopher Moore, and Mark EJ Newman. Random graph models for dynamic networks. *The European Physical Journal B*, 90(10):200, 2017. URL https://dx.doi.org/10.1140/epjb/e2017-80122-8.

[191] Artem Cherkasov, Eugene N. Muratov, Denis Fourches, Alexandre Varnek, Igor I. Baskin, Mark Cronin, John Dearden, Paola Gramatica, Yvonne C. Martin, Roberto Todeschini, Viviana Consonni, Victor E. Kuz'min, Richard Cramer, Romualdo Benigni, Chihae Yang, James Rathman, Lothar Terfloth, Johann Gasteiger, Ann Richard, and Alexander Tropsha. QSAR Modeling: Where Have You Been? Where Are You Going To? *Journal of Medicinal Chemistry*, 57(12):4977–5010, jun 2014. URL http://dx.doi.org/10.1021/jm4004285.

[192] Rudolf Kiralj and Márcia M. C. Ferreira. Basic validation procedures for regression models in QSAR and QSPR studies: Theory and application. *Journal of the Brazilian Chemical Society*, 20(4):770–787, 2009. URL http://dx.doi.org/10.1590/S0103-50532009000400021.

[193] Somayeh Pirhadi, Fereshteh Shiri, and Jahan B. Ghasemi. Multivariate statistical analysis methods in QSAR. *RSC Adv.*, 5(127):104635–104665, 2015. ISSN 2046-2069. URL http://dx.doi.org/10.1039/C5RA10729F.

[194] Lu Xu, Hai-Yan Fu, Qiao-Bo Yin, Yao Fan, Mohammad Goodarzi, and Yuan-Bin She. Interpretable linear and nonlinear quantitative structure-selectivity relationship (QSSR) modeling of a biomimetic catalytic system by particle swarm optimization based sparse regression. *Chemometrics and Intelligent Laboratory Systems*, 159:187–195, dec 2016. URL http://dx.doi.org/10.1016/j.chemolab.2016.10.016.

[195] Frank R. Burden and Dave A. Winkler. Optimal sparse descriptor selection for QSAR using Bayesian methods. *QSAR and Combinatorial Science*, 28 (6-7):645–653, jul 2009. URL http://dx.doi.org/10.1002/qsar.200810173.

[196] Gang Xu and L. G. Papageorgiou. A mixed integer optimisation model for data classification. *Computers & Industrial Engineering*, 56(4):1205–1215, 2009. doi: http://dx.doi.org/10.1016/j.cie.2008.07.012.

[197] Lingjian Yang, Chrysanthi Ainali, Aristotelis Kittas, Frank O. Nestle, Lazaros G. Papageorgiou, and Sophia Tsoka. Pathway-level disease data mining through hyper-box principles. *Mathematical Biosciences*, 260:25–34, 2015. URL http://dx.doi.org/10.1016/j.mbs.2014.09.005.

[198] Jonathan C. Silva, Laura Bennett, Lazaros G. Papageorgiou, and Sophia Tsoka. A mathematical programming approach for sequential clustering of dynamic networks. *The European Physical Journal B*, 89(2):39, feb 2016. ISSN 1434-6028. URL http://dx.doi.org/10.1140/epjb/e2015-60656-5.

[199] George Papadatos, Anna Gaulton, Anne Hersey, and John P. Overington. Activity, assay and target data curation and quality in the ChEMBL database. *Journal of Computer-Aided Molecular Design*, 29(9):885–896, sep 2015. URL http://dx.doi.org/10.1007/s10822-015-9860-5.

[200] Georgia Tsiliki, Cristian R. Munteanu, Jose A. Seoane, Carlos Fernandez-Lozano, Haralambos Sarimveis, and Egon L. Willighagen. RRegrs: An R package for computer-aided model selection with multiple regression models. *Journal of Cheminformatics*, 7(1):1–16, 2015. URL http://dx.doi.org/10.1186/s13321-015-0094-2.

[201] Max Kuhn. *caret: Classification and Regression Training*, 2016. URL https://CRAN.R-project.org/package=caret. R package version 6.0-73.

[202] Harold B Brooks, Sandaruwan Geeganage, Steven D Kahl, Chahrzad Montrose, Sitta Sittampalam, Michelle C Smith, and Jeffrey R Weidner. Basics of enzymatic assays for hts. In *Assay Guidance Manual*. Eli Lilly & Company and the National Center for Advancing Translational Sciences, 2012. URL https://www.ncbi.nlm.nih.gov/books/NBK92007/. Accessed: 2018-01-05.

[203] Rajarshi Guha. Chemical informatics functionality in r. *Journal of Statistical Software*, 18(6), 2007. URL http://dx.doi.org/10.18637/jss.v018.i05.

[204] Kazuhiko Tatemoto. Neuropeptide y: history and overview. In *Neuropeptide Y and related peptides*, pages 1–21. Springer, 2004. URL http://dx.doi.org/10.1007/978-3-642-18764-3_1.

[205] John P Redrobe, Yvan Dumont, and Rémi Quirion. Neuropeptide y (npy) and depression: From animal studies to the human condition. *Life Sciences*, 71(25):2921 – 2937, 2002. ISSN 0024-3205. URL https://dx.doi.org/10.1016/S0024-3205(02)02159-8.

[206] Claes Wahlestedt, Rolf Ekman, and Erik Widerlöv. Neuropeptide y (npy) and the central nervous system: distribution effects and possible relationship to neurological and psychiatric disorders. *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, 13(1):31–54, 1989. URL http://dx.doi.org/10.1016/0278-5846(89)90003-1.

[207] Andrew C Kruse, Jianxin Hu, Albert C Pan, Daniel H Arlow, Daniel M Rosenbaum, Erica Rosemond, Hillary F Green, Tong Liu, Pil Seok Chae, Ron O Dror, David E Shaw, William I Weis, Jürgen Wess, and Brian K. Kobilka. Structure and dynamics of the M3 muscarinic acetylcholine receptor. *Nature*, 482(7386):552–556, feb 2012. URL http://dx.doi.org/10.1038/nature10867.

[208] Kunrong Cheng, Aaron C. Shang, Cinthia B. Drachenberg, Min Zhan, and Jean-Pierre Raufman. Differential expression of M3 muscarinic receptors in progressive colon neoplasia and metastasis. *Oncotarget*, 8(13):21106–21114, feb 2017. URL http://dx.doi.org/10.18632/oncotarget.15500.

[209] Linjun Wang, Xiaofei Zhi, Qun Zhang, Song Wei, Zheng Li, Jianping Zhou, Jianguo Jiang, Yi Zhu, Li Yang, Hao Xu, and Zekuan Xu. Muscarinic receptor M3 mediates cell proliferation induced by acetylcholine and contributes to apoptosis in gastric cancer. *Tumor Biology*, 37(2):2105–2117, feb 2016. URL http://dx.doi.org/10.1007/s13277-015-4011-0.

[210] Fatmah A.M. Al-Omary, Ghada S Hassan, Shahenda M El-Messery, Mahmoud N Nagi, El-Sayed E. Habib, and Hussein I El-Subbagh. Nonclassical antifolates, part 3: Synthesis, biological evaluation and molecular modeling study of some new 2-heteroarylthio-quinazolin-4-ones. *European Journal of Medicinal Chemistry*, 63:33–45, may 2013. URL http://dx.doi.org/10.1016/j.ejmech.2012.12.061.

[211] Joseph H. Chan, Jean S. Hong, Lee F. Kuyper, David P. Baccanari, Suzanne S. Joyner, Robert L. Tansik, Christine M. Boytos, and Sharon K. Rudolph. Selective inhibitors of Candida albicans dihydrofolate reductase: activity and selectivity of 5-(arylthio)-2,4-diaminoquinazolines. *Journal of medicinal chemistry*, 38(18):3608–16, sep 1995. URL http://dx.doi.org/10.1021/jm00018a021.

[212] Mark Whitlow, Andrew J. Howard, David Stewart, Karl D. Hardman, Joseph H. Chan, David P. Baccanari, Robert L. Tansik, Jean S. Hong, and Lee F. Kuyper. X-Ray crystal structures of Candida albicans dihydrofolate reductase: high resolution ternary complexes in which the dihydronicotinamide moiety of NADPH is displaced by an inhibitor. *Journal of medicinal chemistry*, 44(18):2928–32, aug 2001. URL http://dx.doi.org/10.1021/JM0101444.

[213] Asim Kumar Debnath. Pharmacophore mapping of a series of 2,4-diamino-5-deazapteridine inhibitors of Mycobacterium avium complex dihydrofolate reductase. *Journal of Medicinal Chemistry*, 45(1):41–53, 2002. URL http://dx.doi.org/10.1021/jm010360c.

[214] David S. Goodsell. The molecular perspective: methotrexate. *The oncologist*, 4(4):340–1, jul 1999. URL http://dx.doi.org/10.1002/stem.170314.

[215] Aleem Gangjee, Hiteshkumar D Jain, Sherry F Queener, and Roy L Kisliuk. The Effect of 5-Alkyl Modification on the Biological Activity of Pyrrolo[2,3-d]pyrimidine Containing Classical and Nonclassical Antifolates as Inhibitors of Dihydrofolate Reductase and as Antitumor and/or Antiopportunistic Infection Agents(1a-1e). *Journal of Medicinal Chemistry*, 51(15):4589–4600, aug 2008. URL http://dx.doi.org/10.1021/jm800244v.

[216] Lingjian Yang, Songsong Liu, Sophia Tsoka, and Lazaros Georgiou Papageorgiou. Mathematical programming for piecewise linear regression analysis. *Expert Systems with Applications*, 44:156–167, feb 2016. URL http://dx.doi.org/10.1016/j.eswa.2015.08.034.

[217] Alexandros Karatzoglou, Alex Smola, Kurt Hornik, and Achim Zeileis. kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, 11(9):1–20, 2004. URL http://dx.doi.org/10.18637/jss.v011.i09.

[218] Chih-Chung Chang and Chih-Jen Lin. LIBSVM : A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2:1–39, 2013. URL http://dx.doi.org/10.1145/1961189.1961199.

[219] Leo Breiman. Random Forests. *Machine Learning*, 45:5–32, 2001. URL https://dx.doi.org/10.1023/A:1010933404324.

[220] Brian D. Ripley and N. L. Hjort. *Pattern Recognition and Neural Networks*. Cambridge University Press, New York, NY, USA, 1st edition, 1995. ISBN 0521460867.

[221] Lin Song, Peter Langfelder, and Steve Horvath. Random generalized linear model: a highly accurate and interpretable ensemble predictor. *BMC Bioinformatics*, 14(1):5, aug 2013. URL http://dx.doi.org/10.1186/1471-2105-14-5.

[222] Bernard Pirard and Stephen D. Pickett. Classification of kinase inhibitors using bcut descriptors. *Journal of Chemical Information and Computer Sciences*, 40(6):1431–1440, 2000. URL http://dx.doi.org/10.1021/ci000386x.

[223] Peter Csermely, Tamás Korcsmáros, Huba J M Kiss, Gábor London, and Ruth Nussinov. Structure and dynamics of molecular networks: a novel paradigm of drug discovery: a comprehensive review. *Pharmacology & therapeutics*, 138(3):333–408, 2013. URL http://dx.doi.org/10.1016/j.pharmthera.2013.01.016.

[224] Martin Vogt, Dagmar Stumpfe, Gerald M. Maggiora, and Jürgen Bajorath. Lessons learned from the design of chemical space networks and opportunities for new applications. *Journal of Computer-Aided Molecular Design*, 30(3):191–208, mar 2016. URL http://dx.doi.org/10.1007/s10822-016-9906-3.

[225] David Rogers and Mathew Hahn. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, may 2010. URL http://dx.doi.org/10.1021/ci100050t.

[226] Peter Willett, John M Barnard, and Geoffrey M Downs. Chemical Similarity Searching. *Journal of Chemical Information and Computer Sciences*, 38(6): 983–996, 1998. URL http://dx.doi.org/10.1021/ci9800211.

[227] Dilyana Dimova, Dagmar Stumpfe, and Jürgen Bajorath. Quantifying the fingerprint descriptor dependence of structure-activity relationship information on a large scale. *Journal of Chemical Information and Modeling*, 53(9): 2275–2281, 2013. URL http://dx.doi.org/10.1021/ci4004078.

[228] Ye Hu, Dagmar Stumpfe, and Jürgen Bajorath. Advancing the activity cliff concept. *F1000Research*, sep 2013. URL http://dx.doi.org/10.12688/f1000research.2-199.v1.

[229] Gergely Zahoránszky-Kőhalmi, Cristian G. Bologa, and Tudor I. Oprea. Impact of similarity threshold on the topology of molecular similarity networks and clustering outcomes. *Journal of Cheminformatics*, 8(1):16, dec 2016. URL http://dx.doi.org/10.1186/s13321-016-0127-5.

[230] Sofus A. Macskassy and Foster Provost. Classification in Networked Data: A Toolkit and a Univariate Case Study. *Journal of Machine Learning Research*, 8:935–983, 2007. URL http://www.jmlr.org/papers/volume8/macskassy07a/macskassy07a.pdf.

[231] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective Classification in Network Data. *AI Magazine*, 29(3):93, 2008. URL http://dx.doi.org/10.1609/aimag.v29i3.2157.

[232] T. C. Silva and Liang Zhao. Network-Based High Level Data Classification. *IEEE Transactions on Neural Networks and Learning Systems*, 23(6):954–970, jun 2012. URL http://dx.doi.org/10.1109/TNNLS.2012.2195027.

[233] Nagamani Sukumar, Michael P. Krein, Ganesh Prabhu, Sudeepto Bhattacharya, and Subhabrata Sen. Network measures for chemical library design, sep 2014. URL http://dx.doi.org/10.1002/ddr.21218.

[234] Roger J. Griffin, Michelle A. Meek, Carl H. Schwalbe, Malcolm F. G. Stevens, Snc Etoh, and H. N. Nan. Structural studies on bioactive compounds. 8. Synthesis, crystal structure and biological properties of a new series of 2,4-diamino-5-aryl-6-ethylpyrimidine dihydrofolate reductase inhibitors with in vivo activity against a methotrexate-resistant tumor ce. *Journal of Medicinal Chemistry*, 32(13):2468–2474, 1989. URL http://dx.doi.org/10.1021/jm00131a009.

[235] Sherry F Queener. New Drug Developments for Opportunistic Infections in Immunosuppressed Patients: Pneumocystis carnii. *Journal of Medicinal Chemistry*, 38(24):4739–4759, 1995. URL http://dx.doi.org/10.1021/jm00024a001.

[236] Malcolm F. G. Stevens, Keith S. Phillip, Daniel L. Rathbone, Dennis M. O'Shea, Sherry F. Queener, Carl H. Schwalbe, and Peter A. Lambert. Structural Studies on Bioactive Compounds. 28. 1 Selective Activity of Triazenyl-Substituted Pyrimethamine Derivatives against Pneumocystis carinii Dihydrofolate Reductase. *Journal of Medicinal Chemistry*, 40(12): 1886–1893, jun 1997. URL http://dx.doi.org/10.1021/jm970050n.

[237] Claire Robson, Michelle A. Meek, Jan-Dierk Grunwaldt, Peter A. Lambert, Sherry F. Queener, Dirk Schmidt, and Roger J. Griffin. Nonclassical 2,4-Diamino-5-aryl-6-ethylpyrimidine Antifolates: Activity as Inhibitors of Dihydrofolate Reductase from Pneumocystis carinii and Toxoplasma gondii and as Antitumor Agents. *Journal of Medicinal Chemistry*, 40(19): 3040–3048, sep 1997. URL http://dx.doi.org/10.1021/jm970055k.

[238] David C.M. Chan, Charles A. Laughton, Sherry F. Queener, and M. F.G. Stevens. Structural studies on bioactive compounds. 34.1 Design, synthesis, and biological evaluation of triazenyl-substituted pyrimethamine inhibitors of Pneumocystis carinii dihydrofolate reductase. *Journal of Medicinal Chemistry*, 44(16):2555–2564, 2001. URL http://dx.doi.org/10.1021/jm0108698.

[239] Marianne L Richardson, Karen A Croughton, Charles S Matthews, and Malcolm F G Stevens. Structural Studies on Bioactive Compounds. 39. 1 Biological Consequences of the Structural Modification of DHFR-Inhibitory 2, 4-Diamino-6-(4-substituted benzylamino-3-nitrophenyl)-6-ethylpyrimidines ('benzoprims'). *Journal of medicinal chemistry*, 47(16):4105–4108, 2004. URL http://dx.doi.org/10.1021/jm040785+.

[240] Guy W Bemis and Mark A Murcko. The properties of known drugs. 1. Molecular frameworks. *Journal of Medicinal Chemistry*, 39(15):2887–2893, 1996. URL http://dx.doi.org/10.1021/jm9602928.

[241] PubChem-AID1272. Dose response cell-based screening assay for antagonists of neuropeptide y receptor y2 (npy-y2), 5 2008. https://pubchem.ncbi.nlm.nih.gov/bioassay/1279 [Accessed: 07/12/2017].

[242] PubChem-AID1278. Dose response counter screen for neuropeptide y receptor y2 (npy-y2): Cell-based high throughput assay to measure npy-y1 antagonism, 2008. https://pubchem.ncbi.nlm.nih.gov/bioassay/1279 [Accessed: 07/12/2017].

[243] PubChem-AID1279. Dose response counter screen for neuropeptide y receptor y2 (npy-y2): Cell-based high throughput assay to measure npy-y1 antagonism, 2008. https://pubchem.ncbi.nlm.nih.gov/bioassay/1279 [Accessed: 07/12/2017].

[244] PubChem-AID1791. Summary of probe development efforts to identify antagonists of neuropeptide y receptor y2 (npy-y2), 7 2010. https://pubchem.ncbi.nlm.nih.gov/bioassay/1791 [Accessed: 07/12/2017].

[245] David T. Stanton. Evaluation and Use of BCUT Descriptors in QSAR and QSPR Studies. *Journal of Chemical Information and Computer Sciences*, 39(1):11–20, jan 1999. URL http://dx.doi.org/10.1021/ci980102x.

[246] Gisbert Schneider and Uli Fechner. Computer-based de novo design of drug-like molecules. *Nature Reviews Drug Discovery*, 4(8):649–663, 2005. URL http://dx.doi.org/10.1038/nrd1799.

[247] Gisbert Schneider. De novo design – hop(p)ing against hope. *Drug Discovery Today: Technologies*, 10(4):e453–e460, 2013. URL http://dx.doi.org/10.1016/j.ddtec.2012.06.001.

[248] Daniel Reker, Tiago Rodrigues, Petra Schneider, and Gisbert Schneider. Identifying the macromolecular targets of de novo-designed chemical entities through self-organizing map consensus. *Proceedings of the National Academy of Sciences*, 111(11):4067–4072, 2014. URL http://dx.doi.org/10.1073/pnas.1320001111.

[249] Christos A. Nicolaou, Joannis Apostolakis, and Costas S. Pattichis. De Novo Drug Design Using Multiobjective Evolutionary Graphs. *Journal of Chemical Information and Modeling*, 49(2):295–307, feb 2009. URL http://dx.doi.org/10.1021/ci800308h.

[250] R. Vasundhara Devi, S. Siva Sathya, and Mohane Selvaraj Coumar. Evolutionary algorithms for de novo drug design – a survey. *Applied Soft Computing*, 27:543 – 552, 2015. ISSN 1568-4946. URL https://dx.doi.org/10.1016/j.asoc.2014.09.042.